GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION
SPONSORED PROJECT INITIATION

No action

Date: January 5, 1978

Project Title: The Development of a Computer Game for Teaching Management Information Systems

Project No: M-50-625

Project Director: James F. Courtney, Jr.

Sponsor: Exxon Education Foundation

Agreement Period: From 10/25/77 Until 10/24/78

Type Agreement: Letter dated 10/25/77

Amount:
$22,250 Exxon
12,990 GIT (M-50-317)
$35,240 Total

Reports Required: Semi-Annual Progress Reports

Sponsor Contact Person (s):

Technical Matters

Contractual Matters
(thru OCA)

Miss Barbara Wright
Project Officer
Exxon Education Foundation
111 West 49th Street
New York, New York 10020
(212) 398-2273

Defense Priority Rating: none

Assigned to: Industrial Management EES (School/Laboratory)

COPIES TO:

Project Director
Division Chief (EES)
School/Laboratory Director
Dean/Director—EES
Accounting Office
Procurement Office
Security Coordinator (OCA)
Reports Coordinator (OCA)
Library, Technical Reports Section
EES Information Office
EES Reports & Procedures
Project File (OCA)
Project Code (GTRI)
Other

CA-3 (3/76)

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION

Date: March 27, 1980

Project Title: The Development of a Computer Game for Teaching Management Information Systems

Project No: M-50-625

Project Director: James F. Courtney, Jr.

Sponsor: Exxon Education Foundation

Effective Termination Date: August 31, 1979

Clearance of Accounting Charges: August 31, 1979

Grant/Contract Closeout Actions Remaining:

- x Final Invoice and Closing Documents
- _ Final Fiscal Report
- _ Final Report of Inventions
- _ Govt. Property Inventory & Related Certificate
- _ Classified Material Certificate
- _ Other _____

Assigned to: Industrial Management (School/Laboratory)

COPIES TO:

Project Director
Division Chief (EES)
School/Laboratory Director
Dean/Director—EES
Accounting Office
Procurement Office
Security Coordinator (OCA)
Reports Coordinator (OCA)

Library, Technical Reports Section
EES Information Office
Project File (OCA)
Project Code (GTRI)
Other C. E. Smith

CA-4 (1/79)

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION
SPONSORED PROJECT INITIATION

Date: January 5, 1978

Project Title: The Development of A Computer Game for Teaching Management Information Systems

Project No: B-499 (Sub-Project under M-50-625/Courtney/IM)

Project Director: Frank Vogler

Sponsor: Exxon Education Foundation

Agreement Period: From 10/25/77 Until 10/24/78

Type Agreement: Letter dated 10/25/77

Amount: $10,248

Reports Required: Semi-Annual Progress Reports

Sponsor Contact Person (s):

Technical Matters	Contractual Matters (thru OCA)
Miss Barbara Wright Project Officer Exxon Education Foundation 111 West 49th Street New York, New York 10020 (212) 398-2273	

Defense Priority Rating: none

Assigned to: SED/EES (School/Laboratory)

COPIES TO:

- Project Director
- Division Chief (EES)
- School/Laboratory Director
- Dean/Director–EES
- Accounting Office
- Procurement Office
- Security Coordinator (OCA)
- Reports Coordinator (OCA)
- Library, Technical Reports Section
- EES Information Office
- EES Reports & Procedures
- Project File (OCA)
- Project Code (GTRI)
- Other

CA–3 (3/76)

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION

Date: March 27, 1980

Project Title: The Development of A Computer Game for Teaching Management Information Systems

Project No: B-499 (Sub project under M-50-625/Courtney/IM)

Project Director: Frank Vogler

Sponsor: Exxon Education Foundation

Effective Termination Date: August 31, 1979

Clearance of Accounting Charges: August 31, 1979

Grant/Contract Closeout Actions Remaining: NONE

_ Final Invoice and Closing Documents
_ Final Fiscal Report
_ Final Report of Inventions
_ Govt. Property Inventory & Related Certificate
_ Classified Material Certificate
_ Other _____

Assigned to: EES/SEL (School/Laboratory)

COPIES TO:

Project Director
Division Chief (EES)
School/Laboratory Director
Dean/Director–EES
Accounting Office
Procurement Office
Security Coordinator (OCA)
✓ Reports Coordinator (OCA)
Library, Technical Reports Section
EES Information Office
Project File (OCA)
Project Code (GTRI)
Other _____

CA–4 (1/79)

M5062

Progress Report
The Development of a Computer Game for Teaching
Management Information Systems

Georgia Institute of Technology
April 19, 1978

1. Progress to date has been excellent, but see 2.
2. Funds for the project were received in October; however, we were in the midst of the fall quarter at the time and could not begin work until January. Thus, the schedule may have to be moved back some. This should not require any change in funding.
3. No changes in personnel. Virginia Breedlove is the graduate assistant. Frank Vogler the analyst.
4. No major problems.
5. Funds appear to be adequate.

EXXON EDUCATION FOUNDATION

REPORT PERIOD: FROM October 25, 1977 TO March 31, 1978

NT RECIPIENT	PROJECT DIRECTOR
Georgia Institute of Technology	James F. Courtney, Jr.

JECT TITLE
Development of a Computer Game for Teaching Management Information Systems

AL AWARD	PAYMENTS TO DATE				TOTAL PAYMENTS
22,250		$11,125			$11,125

EXPENDITURE		PREVIOUSLY REPORTED	CURRENT HALF-YEAR	TOTAL
LARIES	PROFESSIONAL	-	$1453.60	$1453.60
	CLERICAL	-	-	-
	STUDENT	-	429.00	429.00
	OTHER	-	-	-
PLOYEE BENEFITS			85.83	85.83
NSULTING FEES				
AVEL EXPENSES				
UIPMENT			1158.00	1158.00
PPLIES				
BLICATION				
TOTALS		-	3126.43	3126.43

PENDED BALANCE	DATE OF REPORT
$7998.57	April 19, 1978

EASE YPE	NAME OF CHIEF BUSINESS OFFICER	TITLE
	NAME OF PERSON PREPARING REPORT: James F. Courtney, Jr.	TITLE: Assistant Professor

SIGNATURE OF ABOVE BUSINESS OFFICER

SIGNATURE OF ABOVE PROJECT DIRECTOR

11-SC-625

EXXON EDUCATION FOUNDATION

REPORT PERIOD: FROM April 1, 1978 TO November 30, 1978

GRANT RECIPIENT	PROJECT DIRECTOR
Georgia Institute of Technology	James F. Courtney, Jr.

PROJECT TITLE
Development of a Computer Game In Teach MIS

TOTAL AWARD	PAYMENTS TO DATE				TOTAL PAYMENTS
22,500		11,250	11,250		22,500

EXPENDITURE		PREVIOUSLY REPORTED	CURRENT HALF-YEAR	TOTAL
SALARIES	PROFESSIONAL	1453.60	13517.38	14970.98
	CLERICAL			
	STUDENT	429.00	1000.00	1429.00
	OTHER			
EMPLOYEE BENEFITS		85.83	725.49	811.32
CONSULTING FEES				
TRAVEL EXPENSES			901.28	901.28
EQUIPMENT		1158.00		1158.00
SUPPLIES			98.95	98.95
PUBLICATION				
TOTALS		3126.43	16243.10	19369.53

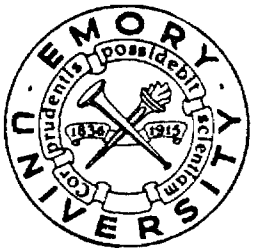
UNEXPENDED BALANCE	DATE OF REPORT
3130.47	December 20, 1978

LEASE TYPE	NAME OF CHIEF BUSINESS OFFICER	TITLE
	NAME OF PERSON PREPARING REPORT	TITLE
	James F. Courtney, Jr.	Assistant Professor

SIGNATURE OF ABOVE BUSINESS OFFICER

SIGNATURE OF ABOVE PROJECT DIRECTOR

Comments: Progress satisfactory
No major problems or personnel changes



Final Report
To EXXON Education Foundation

Development of a Computer Game for Teaching Management Information Systems

James F. Courtney, Jr.
Texas Tech University

Ronald L. Jensen
Emory University

August 1979

COMPUTATION LABORATORY
SCHOOL OF BUSINESS ADMINISTRATION
EMORY UNIVERSITY
Atlanta, Georgia 30322

Table of Contents

FINAL REPORT
DEVELOPMENT OF A COMPUTER GAME FOR TEACHING MANAGEMENT INFORMATION SYSTEMS

Executive Summary

The project to develop a computer game for use in management information systems classes has been highly successful. The basic purpose of the project was to provide students who are unsophisticated computer users with the tools necessary to develop management information systems. This has been done by developing a student-accessible data base and a simple query language to be used in conjunction with an existing business game, The Business Management Laboratory (BML).

The new system is called the Systems Laboratory for Information Management (SLIM). Classroom use indicates that the BML/SLIM package is an extremely useful educational tool. Students find that playing BML is challenging and enjoyable. The semi-structured and somewhat uncertain environment provided by BML also motivates the need for a management information system to assist in decision-making. The SLIM system provides the data base and tools necessary for developing the MIS.

Evaluation of the BML/SLIM Package

The BML/SLIM package has been used in two classes at Emory and two at Georgia Tech. The instructors have been extremely pleased with the results.

Course evaluation questionnaires indicate that student reaction is very favorable towards this approach to teaching MIS. For example, at Georgia Tech over three-fourths of the students in one class indicated that the BML/SLIM approach was equal or superior to a standard lecture/case method.

At Emory, students using SLIM were asked to rate its usefulness on a scale from 1 to 10. The mean rating was 7. Over 40% rated its usefulness as 9 or 10. One student on the only team that chose not to use SLIM gave it a rating of 7 based on what he heard from other teams that did use it.

While the results of these questionnaires are most encouraging, they must be considered somewhat tentative because the conditions under which they were administered were not sufficiently well controlled for formal experimental analysis. More definitive conclusions must await the development of controlled classroom experiments and data collection over a two or three year period.

On the technical side, results have also been highly satisfactory. The extension of time permitted not only full compliance with the original project objectives, but even allowed our going beyond original intentions in some areas. Most notably, the additional time made it possible to include development of the commands REMARK, WHEN, BATCH, REWIND, and TERMINAL resulting in a much more powerful query system.

As far as the objective of ease-of-use is concerned, the questionnaires showed that even nontechnically-oriented students were able to master the query language quite easily even though only two to three lecture hours were devoted to it. As mentioned previously, students also rated the system as being very useful. Additional evidence of this was a 640 line canned SLIM MIS developed by one team. The team claimed that this MIS reduced decision-making time by 40%.

Regarding size and portability, the SLIM program consists of about 2500 FORTRAN statements (including many comments). Care has been taken during programming to insure that the code is within the ANSI X3.9-1966 standards for portability. Although this does not totally insure portability, we feel confident in stating that the system can be easily installed on a minicomputer with 64k words of main memory and an ANSI FORTRAN compiler. The system has been used on a 32k mini by using overlays.

The system has been successfully installed on the following machines:

1) PDP-8 with the OS/8 operating system

2) CDC CYBER 74 with the NOS operating system

3) UNIVAC 90/80 with the VS/9 operating system

Installation of the system on three different machines lends credence to the notion of portability. In addition, Professor Richard Scammell on the management faculty at the University of Houston is currently conducting a trial installation and use of the system. This is being done without assistance from the project staff in order to test the documentation as well as portability. The system will be used in his class this fall.

Dissemination of Project Results

We are most happy to report that existence of the BML/SLIM system has already been widely publicized and response from the academic community has been very encouraging. BML/SLIM was featured in the March, 1979 issue of the "Computing Newsletter for Schools of Business" published by Daniel Couger at the University of Colorado at Colorado Springs. Couger also discussed BML/SLIM in a recent article in "Decision Line", a newsletter published by the American Institute for Decision Sciences (AIDS).

Also, as a result of this project Professor Courtney was asked to serve on a panel discussing MIS education at the Tenth Annual Conference of the Southwest Chapter of AIDS. Professor Courtney discussed the BML/SLIM approach.

Five papers on various aspects of BML/SLIM have already been presented at professional meetings and two others are under preparation for presentation this Fall. These papers are:

"Security in a Shared Data Base for a Management Information System Game," presented at the Tenth Annual Meeting of the Southwest Chapter of AIDS, March 1979.

"Incorporating MIS/DSS into Policy Courses via Simulation," presented at the Sixth Annual Conference of the Association for Business Simulation and Experiential Learning, April 1979.

"An Instructional Decision Support System," presented at DECUS, the Digital Equipment Corporation User's Society meeting, April 1979.

"An Instructional System for Integrating Management Models and Generalized Data Base Management Systems," presented at the Simulation Management Workshop sponsored by the Army Institute for Research in Management Information and Computer Science, May 1979.

"Incorporating Management Policy and Strategy into MIS Courses via an MIS Game," presented at the TIMS International Meeting, June 1979.

"A Management Simulation for Teaching MIS and DSS," to be presented at the 1979 National Meeting of AIDS, October 1979.

"An Instructional Package for Decision Support Systems," to be presented at the Eighteenth Annual Meeting of the North American Simulation and Gaming Association, October 1979.

Copies of these papers are available from the authors.

As a result of the dissemination of information about the project, enquiries about BML/SLIM have already been received from over 15 instutions.

Conclusions

Overall we have been very pleased with the results of the project. We believe that the original objectives have at least been met and in some instances exceeded. We have have been very satisfied with use of the system in class as feedback from students has been favorable. Response from the academic community has also been quite good. The real test of the system, however, will come from its widespread use in the field. Unfortunately it will be two or three years before this can take place and enough data is available to fully evaluate the system.

****
* *
* S L I M *
* *
* *
* System Laboratory for *
* Information Management *
* *
* Business Management Laboratory *
* Data Management System *
* *
****

****

User Manual

****

by

James F. Courtney, Jr.
Texas Tech University

and

Ronald L. Jensen
Emory University

Preliminary Version
Rev. August 1979

To all of the students who
have worked so hard to
understand computers and their
managerial implications
and
to all of the instructors who
have worked to provide
this understanding.

PREFACE

This manual provides the user instructions for a data base query language written to accompany The Business Management Laboratory management simulation. It is intended to illustrate many of the concepts of data base management, management information systems, and decision support systems. The development of this program was supported by a grant from the Exxon Education Foundation, to whom we are extremely gratefull.

This is a pilot version of the user instructions, and so the authors would appreciate any feed-back from the users on its accuracy and completeness.

Development of the SLIM program was the work of several persons, to whom proper recognition needs to be given. In particular, the contributions of Frank Vogler, Gordon Rapkin, Ginger Breedlove, and David Paradice need to be acknowledged. While SLIM received the benefit of input from several persons, the authors accept the responsibility for the errors or ommissions.

[The SLIM program is still considered to be in a testing phase, and while the authors accept responsibility for correcting errors, no warranty is made as to its accuracy or completeness.]

Ronald L. Jensen
Atlanta, GA

James F. Courtney
Lubbock, TX

TABLE OF CONTENTS

System Laboratory for
Information Management

Business Management Laboratory
Data Management System

User Manual

1. Introduction

As a participant in the Business Management Laboratory, you have the option of purchasing a data management system (SLIM) that can be used to build an information system to assist you in making decisions. A well designed information system should help you make better decisions more easily and more confidently.

SLIM is a software system that has been developed for illustrating concepts of data management. It is similar to many commercial packages in use today. The components of the system are a data base with items derived from play of the BML game, and a data base query language (DBQL) for accessing the data. SLIM creates the data base and updates it as play of the game proceeds.

The data base consists of values for 126 different items for each quarter of play. Example data items are product prices, raw material prices, advertising expenditures, sales, cost-of-goods sold and net income. The data base is discussed fully in Section 2 and a list of data items is given in Table 2.1.

The data base query language can be mastered in a few hours and has been designed to allow users who may be uninitiated in computer systems to access the data base. The DBQL consists of commands that allow you to retrieve and print data (or store it on another file), make computations based on the data, and to prepare reports for use in decision-making. The DBQL is discussed in Section 3.

The purchase cost of SLIM is $4000. This includes the DBQL, all data for your firm and industry and certain data for other firms. Each quarter your firm will be assessed a usage charge of $500.00 for computer time, supplies, etc.

2. The Data Base

A complete list of items in the SLIM data base is given in Table 2.1. The first column, labeled DATA ITEM, indicates the title that you will use for that data item in DBQL statements. For example, "PRInt BBI;" will retrieve and print-out the current value of the Bull-Bear Stock Index.

Items in the data base may also be accessed with subscripts. For example, if you want the Bull-Bear Index for the eighth quarter of play (assuming at least 8 quarters have been played), you would enter "PRInt BBI(Q8);". Other aspects of using subscripts are discussed in Section 3.1.3.

Also, many items for firms other than your own are not available immediately after a quarter is played, but are delayed somewhat. The column labeled "Delay" indicates the number of quarters that these data items lag. A delay of 99 quarters indicates that the item is never available to your firm. For example, if 12 quarters have been played, the most recent value of accounts receivable that you can obtain for any other firm is for quarter 10. You may never access lost sales resulting from stock-outs for another firm because its delay is 99.

Finally, values for certain data items of other firms will not be known precisely, but only within some percentage of error. This is indicated in the column labeled "Percent Error." For example, if you retrieve the cost-of-goods sold for another firm and the true value is $100,000, the figure you get may range from $90,000 to $110,000 because of a ten percent error. This simulates the data measurement error problem which you would encounter in trying to obtain this information in an actual situation. Any figure in this range is equally likely; however, because of the manner in which the randomization is done, you will get the same value each time the item is retrieved.

Table 2.1
Data Dictionary

The following items are available in the BML-SLIM data base. They are replicated, where necessary, for each firm for each period, with access controlled by individual firm passwords.

DATA ITEM		DELAY	%ERROR
Environmental			
BBI	Bull-Bear Index (stock market)	0	00
EI	Current Economic Indicator	0	00
EINQ	Economic Indicator Forecast, next period	0	00
EINY	Economic Indicator Forecast, next year	0	00
FUT	Current Period Futures Quotation (material)	0	00
PRAW	Current Period Spot Quotation (material)	0	00
QTR	The current (most recent) quarter	0	00
STBR	Current Period Bill Rate (e.g. T-bills)	0	00
YEAR	The current (most recent) year	0	00
Finance and Administration			
AC	Administrative Costs	1	10
AP	Accounts Payable	1	10
AR	Accounts Receivable	1	10
ARF	Accounts Receivable to be Factored	99	00
BL	Net Earnings (Bottom Line) after taxes	0	00
BOUT	Outstanding Bonds	0	00
CASH	Cash Assets	1	10
COGS	Cost of Goods Sold	1	10
CR	Credit Rating	99	00
DIVD	Dividends Declared, current period	0	00
FC	Fixed Costs	99	00
FE	Factoring Expense	1	10
FVBO	Change in FaceValue of Bonds (sale or call)	0	00
IE	Interest Expense	1	10
IT	Type of Short-Term Investment	99	00
ME	Miscellaneous Expenses	1	10
OE	Owners Equity	1	00
PIC	Paid-in Capital	1	00
SALES	Sales Revenue	1	00
SOFF	Number of Shares Stock Offerred	0	00

SOUT	Number of Shares Stock Outstanding	0	00
SP	Stock Price, last closing	0	00
SPL	Outstanding Special Loans	99	00
SSP	Stock Sales Price, new issue	0	00
SSR	Stock Sales Receipts, new issue	0	00
STI	Short Term (90 day) Investment, amount	99	00
STII	Short Term Investment Income	0	00
STIR	Short Term Investment, rate	99	00
STL	Short Term Loan	1	00
STLO	Short Term Loans Outstanding	1	00
TAX	Tax Liability	1	00
TL	Term Loan Balance	1	00
TLIR	Term Loan Interest Rate (quarterly)	99	00
UBD	Unamortized Bond Discount	0	00

Marketing and Sales

ADV	Advertising Budget (prod, area)	1	00
BO	Backorders (prod, area)	99	00
COMM	Sales Commission (prod)	1	00
LS	Lost Sales to to Stock Out (prod, area)	99	00
MS	Market Share (prod, area)	1	00
PRICE	Price (prod, area)	0	00
QC	Quality Control Budget	1	10
QC2%	% of QC Budget which goes to product 2.	1	20
SAL	Salary for Sales Representatives	1	10
SC	Total Sales Compensation, current	1	10
SHIP	Flag to indicate shipping policy	99	00
SREPS	Number of Sales Representatives (by area)	0	00
SV	Sales Volume, units (prod, area)	1	10
TRAIN	Sales Trainees, current	99	00
UTX	Units Transshipped (prod, area)	99	00

Plant and Production

CE	Capital Expenditures, current	99	00
CPC	Change in Production Capacity (stage, area)	1	20
DEPR	Depreciation Charge	1	00
FGI	Finished Goods Inventory (prod, area)	1	10
FGUC	Finished Goods Unit Cost (prod, area)	1	10
ICC	Inventory Carrying Charges	1	10
MAIN	Plant Maintenance Expenditures (stage, area)	1	20

PC	Plant Capacity in labor-hours (stage, area)	1	20
PLAN	Book Value of Plant	1	20
PSB	Total Research Budget	1	20
PVFS	Production Volume, first shift (prod, area)	1	20
PVSS	production Volume, second shift (prod, area)	1	20
RMI	Raw Material Inventory (mat, area)	1	20
RMO	Raw Material Orders (mat, area)	99	00
RMUC	Raw Material Unit Cost (mat, area)	1	25
SS%	Second Shift Production Available, r.e. first shift (area)	99	00
TPC	Total Production Costs (prod, area)	1	20

3. The Data Base Query Language

In order to use SLIM, you must first log-in using the procedure for your computer installation. Since this procedure varies considerably among installations, your instructor will explain it to you. However, once you have logged-in and invoked the SLIM system, you will be asked to identify your firm and its associated password (your instructor will give you the password for your firm). Be sure to maintain the secrecy of your password or other teams may gain access to your private data.

3.1 Starting and Ending SLIM

When SLIM is started, it greets you and asks for your firm number. You respond with your one digit firm number and a return. SLIM will then ask for your password. On some systems SLIM will "black-out" an area for you to type your password to provde some sequrity. You then enter the user number and password for your firm and return the carriage. Your user number will be a two digit number given to you by your instructor. The system will print an ellipsis (...), which means it is ready to accept SLIM commands. Some systems may also print a prompt character such as a ? or *. If your number or password is entered incorrectly, SLIM will ask for your firm number and password again. If you don't get this right after three attempts, SLIM issues an insulting message and rides off into the sunset.

When you type in a command, it is immediately processed by SLIM and the results are returned to your terminal. Another command is then solicited through the printing of another ellipsis. Each command, except a REMark discussed below, must be terminated by a semicolon (;). If you inadvertantly omit the semicolon, simply type it on the next line. More than one statement can be entered on one line or a statement can overlap lines, if necessary. A command can be too long for SLIM to process. If so you will get error number "30". If this happens, just split the query into two or more smaller commands. [Note: Do not type more than 70 characters on a line or the final characters will be lost by the system. It is better to type multiple short lines which do not risk exceeding the 70 character limit.] Whenever you have completed entering a line, always return the carriage. Blanks are permissable anywhere except within data item names or within the SLIM command names discussed below. Any command can be abbreviated to its first 3 characters. (To emphasize this the first 3 characters of each command are in upper case throughout the manual.) The query will always begin with a command name (for example PRInt). At least one blank must

separate the command name from the rest of the query.

When you are ready to terminate SLIM, you simply enter

END;

If you are also done with your session at the terminal, follow the log-out procedures for your installation.

3.2 Retrieval Commands

Two commands are available for retrieving data and having it printed at your terminal. These are the PRInt and LISt commands. The SUBset command retrieves data, but writes it to a temporary file. This is discussed in section 3.2.5.

3.2.1 PRInt

The PRInt and LISt commands are similar, except that PRInt formats output into a single vertical column with labeling on the left side, whereas LISt formats output (up to 6 data values) into columns with headings at the top of each column. No more than 10 data item names can be used in a single query and no more than 128 data values can be retrieved per query.

A PRInt statement has the general form:

PRInt <item list>;

where <item list> is a list of names of data base items, temporary variables (discussed in Section 3.2.5.1), or functions (Section 3.2.6) separated by commas and ending with a semicolon.

Examples:

To print cash-on-hand, short-term investments and credit rating for your firm in the current quarter (unless otherwise stated, assume firm 1, quarter 4):

PRInt CASH, STI, CR;

CASH(F1,Q4) = 10010.00
STI(F1,Q4) = 25000.00
CR(F1,Q4) = 5.00.

3.2.2 Subscripts

The retrieval of certain data base items requires the use of subscripts (indexes) to indicate precisely which item is desired. For example, to retrieve product prices or sales volume, you must indicate both the product and market area. Subscripts follow the item name and are enclosed in parentheses. The sequence in which subscripts are entered is immaterial [see section 4 for notes on optimized usage of SLIM.]. Data base items which must be subscripted are indicated in parentheses in the Description column of Table 2.1. Although not shown, any item may be subscripted by quarter. The initial quarter of data is denoted Q0 (Q zero). Data from the first set of student decisions is Q1. Also, any item for a firm (everything but Environmental factors) can be subscripted by firm.

3.2.2.1 Abbreviations

The following abbreviations are used in Table 2.1 and in subscripts:

Table 3.1
Subscript Symbols

FACTOR	SYMBOL	DEFAULT
Area	A	1
Firm	F	Your firm
Material	M	1
Product	P	1
Quarter	Q	Current
Stage	S	1

If F (firm) and Q (quarter) are omitted, data values for your firm in the most recent quarter are implied. Other subscripts default to a value of 1 when omitted.

Examples:

1) Print product prices for your firm in the most recent quarter for product 1, area 2.

```
PRInt PRICE(P1,A2);

PRIC(F1,Q4,P1,A2) = 41.00
```

Print plant capacity for stage 1, area 1 for your firm in the most recent quarter.

```
PRInt PC(A1,S1);

PC(F1,Q4,A1,S1) = 9000.00
```

2) Print advertising budget and sales volume for firm 2, product 1, area 1, quarter 8.

```
PRI ADV(F2,P1,A1,Q8), SV(A1,Q8,F2,P1);

ADV(F2,P1,A1,Q8) = 2000.
SV(A1,Q8,F2,P1) = 1500.
```

If you want a range of values for a subscripted data item, then you may simply enter, in order, the index code, lower value, a dash, upper value. If more than one index has a range, the one to the right will vary most rapidly. The index second from right will vary next most rapidly, etc. For clarification, see example 2 below.

Examples:

1) Print the Bull-Bear Index for quarters 1 to 4.

```
PRI BBI(Q1-4);

BBI(Q1) = 98.25
BBI(Q2) = 101.40
BBI(Q3) = 62.15
BBI(Q4) = 151.32
```

2) Print finished goods inventory and backorders for firms 1 and 2, product 1, quarters 7 to 9, area 1.

```
PRI FGI(Q7-9,F1-2,P1,A1), BO(Q7-9,F1-2,P1,A1);

FGI(Q7,F1,P1,A1) = 7000.00
FGI(Q7,F2,P1,A1) = 2000.00
FGI(Q8,F1,P1,A1) = 6000.00
FGI(Q8,F2,P1,A1) = 3500.00
FGI(Q9,F1,P1,A1) = 2000.00
FGI(Q9,F2,P1,A1) = 3000.00
BO(Q7,F1,P1,A1) = 0.00
BO(Q7,F2,P1,A1) = 0.00
BO(Q8,F1,P1,A1) = 0.00
BO(Q8,F2,P1,A1) = 0.00
BO(Q9,F1,P1,A1) = 0.00
BO(Q9,F2,P1,A1) = 0.00
```

3.2.2.2 Wild Cards (*)

The asterisk (*) may be used within the subscript as a substitute for the last value in a range. This permits generalized commands, which may be particularly useful in preparing a type of "standard analysis". For example, the subscript F1-* would indicate "all firms", and Q1-* would indicate "all quarters". Thus the command:

```
PRI BBI(Q1-*);
```

would print the Bull-Bear Index for all periods to date.

When the wild card is used with other firms, you get all the information which passes the delay test, but all data within the delay period is zeroed. Thus, if there are 8 periods in the data base and you ask for the information over the range Q0-*, and the data has a one period delay, the first seven data items would be reported and the last item would be shown as zero. A reminder message is printed at the terminal. The side effects of this are:

a. You don't get an error message if you request delayed data, but SLIM gives you a warning message and prints a value of zero.
b. It is possible to LISt data for your own firm and other firms by quarter without an "unequal column length" error. SLIM will give you all of the information for your own firm, all of the allowed information for the other firms, and then fills out the table with zeros to permit the listing.

3.2.3 LISt

The LISt command formats output into columns. Only data item names and temporary variables may appear in a LISt command. No more than six data items and temporary variables can be LISted in one command. A column is created for each data item and each temporary variable in the command. The columns are in the same sequence as the items in the LISt command. Remember that the leftmost subscripts vary the slowest. Finally, each column must be the same length. Be sure to check that subscripts conform to this restriction. The first line of LISt output consists of data item names. The second line consists of the subscript elements which do not vary over a range. It will usually be convenient to have only one subscript vary. Examples:

1) LISt prices for products 1 and 2 in market area 1 for quarters 1-4 for your firm. (Output is in ascending order by quarter.)

```
LIS PRIC(P1,Q1-4), PRIC(P2,Q1-4);
```

PRIC F1,P1,A1	PRIC F1,P2,A1
39.50	9.00
39.00	9.25
41.50	9.50
41.50	9.50

2) LISt accounts receivable in order by quarter preparing a column for firms 1,3,2, quarter 3-5.

```
LIS AR(F1,Q3-5), AR(F3,Q3-5), AR(F2,Q3-5);
```

AR F1	AR F3	AR F2
50000.00	40000.00	38000.00
65000.00	50000.00	40000.00
70000.00	70000.00	66600.00

The first column above corresponds to firm 1, the second to firm 3, and the third to firm 2.

3.2.4 OPTion: Totaling and Averaging Columns in LISt Commands

Special options are available for either totalling each column in a LISt command or averaging each column (but not both simultaneously). This is done by using the OPTion command. The command OPT 1; will result in totals being printed at the bottom of each column in subsequent LISt commands. OPT 2; will produce averages for each column.

Once an option is invoked, it remains in effect until it is changed with a new option command or turned off with an OPT 0; command.

3.2.5 SUBset

Instead of writing the results of retrievals at your terminal, SUBset writes onto a temporary output file. This file is given a standard name for your system. Your instructor will give you information on the name of the file and any other system dependent information. The same restrictions that apply to LISt also apply to SUBset, except that up to 10 values may be output per command. However, no more than 6 values will be put on each line. If you have 6 or fewer data items or temporary variables in the SUBset command, each line will contain values for exactly the number of data items or temporary variables you have named. If you have more than 6, two lines of output will be required.

Each data value will have a maximum of 12 characters, including a decimal point. Two decimal positions are always included. The file is written with a FORTRAN format statement of F12.2 for absolute values greater than or equal to 10.0, or F12.4 for absolute values less than 10.0. This file could then be read by a FORTRAN program using 6F12.2 as the input format.

Examples:

1) On the temporary file, put product 1 sales volume, advertising, and price for quarters 1-3 all for your firm in area 1.

```
SUB SV(Q1-3), ADV(Q1-3), PRIC(Q1-3);
```

The result of this command would be three columns of output, one for SV, ADV, and PRIC, respectively. Each column would have three lines in sequence by quarter.

2) On the temporary file, put stock price, credit rating, sales revenue , net earnings, owner's equity, book value of plant, paid in capital, and dividends for your firm for quarters 1-6.

```
SUB SP(Q1-6),CR(Q1-6),SALE(Q1-6),BL(Q1-6),
OE(Q1-6),PLANT(Q1-6), PIC(Q1-6), DIVD(Q1-6);
```

This command would result in a total of twelve lines being written to the output file. Odd numbered lines would have six values corresponding to SP, CR, SALES, BL, OE, and PLANT, respectively. Even numbered lines would contain values for PIC and DIVD.

3.2.6 COMpute

The COMpute command allows you to calculate values for what are called temporary variables. For example, the COMpute statement below defines a value of 4 for the temporary variable A.

```
COMpute A = 2 + 2;
```

A COMpute statement will always have an equal sign with a temporary variable name to the left and an arithmetic expression to the right. That is, the general form of this statement is:

```
COM <temporary variable> = <arithmetic expression>;
```

3.2.6.1 Temporary Variables

Temporary variables have names which consist of a single alphabetic character. Variables X, Y, and Z are arrays and are used in conjunction with subscripts. Variables A through W may not have subscripts. Subscripts for temporary variables must be constants, except as discussed in the secton on indirect subscripting. For example, X(3) refers to the third element of the array X. Arrays have a maximum of 16 elements. Unlike retrieval commands, subscripts in COMpute commands may not have a range of values (except in conjunction with functions discussed below). Temporary varables are initialized to a value of zero.

3.2.6.2 Arithmetic Expressions

Arithmetic expressions are portions of statements which consist of logical arrangements of data base items, constants, temporary variables (which have already appeared to the left of an equal sign in a previous COMpute statement), functions, and arithmetic operators. Arithmetic operators are defined in Table 3.2. In addition to the limit of 10 data item names, no more than 10 constants can be used in a COMpute and no more than 10 functions are allowed.

TABLE 3.2
Arithmetic Operators

Operation	Operator	Priority
Addition	+	1
Subtraction	-	1
Multiplication	*	2
Division	/	2
Exponentiation	**	1

Thus if you want temporary variable S to be the sum of cash, short-term investments, and accounts receivble for your firm in the current quarter, the appropriate statement would be:

```
COMpute S = CASH + STI + AR;
```

If you next decided to define temporary variable R as S divided by short-term loans, the statement would be:

```
COM R = S/STL;
```

If you want to know the value of R, you must PRInt or LISt it.

3.2.6.3 Hierarchy of Expression Evaluation

When SLIM evaluates an expression, a certain hierarchy is applied to the operators. You may think of the evaluation procedure as a process consisting of three passes through the COMpute statement. The passes are from left-to-right through the statement. Terms consisting of exponentiation are done on the first pass (left-most first). Multiplication and division are done on the second pass. Finally, addition and subtraction are done on the last pass. Each time, terms to the left are evaluated first.

For example, the statement

```
COM B = 1 + 2**2 - 4/2 + 3 * 4**2;
```

would be evaluated as follows:

First pass:

1. 2**2 = 4 is done first

2. 4**2 = 16 is done second

Producing: 1+4-4/2+3*16

Second pass:

3. 4/2 = 2 is done first

4. 3 times the result of 4**2 is done second to give 48

Producing: 1+4-2+48

Third pass:

5. The additions or subtractions are executed,
Producing: 51 as the final answer.

Thus B will have a value of 51.

3.2.6.4 Using Parentheses to Alter the Hierarchy of Evaluation

You may alter the sequence of expression evaluation through the use of parentheses. For example, suppose you wanted to divide the sum of your firm's cash and accounts receivable by the short-term loans for the current quarter. A statement of the form COM A = CASH + AR / STL; would not work because accounts receivable would be divided by short-term loans and the result would be added to cash. One way to correct this is by using two statements:

```
COM A = CASH + AR;  COM B = A/STL;
```

However it is more convenient to use parentheses to set-off the addition operation. The statement would be

```
COM A = (CASH + AR) /STL;
```

The expression in parentheses is evaluated first. Within parentheses the hierarchy discussed previously is used. You may use two levels of parentheses (not including subscripts).

Examples:

1. Express lost sales of product 1 in quarter 5 as a percent of sales of product 1 for your firm in area 1.

```
COM  T = (LS(Q5,P1,A1) * PRICE(Q5,P1,A1)) / (SV(Q5,P1,A1)
* PRICE(Q5,P1,A1)) * 100;  PRI T;
```

2. Calculate the "quick ratio" for your firm for this quarter.

```
COM C = (CASH + AR + STI) / (AP + STL + SPL)

PRI C;
```

3.2.7 Functions

Some predefined statistical functions are available for use in PRInt and COMpute commands. Included are functions to find the SUM, MINimum, MAXimum, AVErage or LOG(base e) of items specified in an argument list. The argument list consists of data item names, temporary variables, and constants. The argument list is enclosed in parentheses. When functions are specified in a

COMpute command, they are evaluated before any arithmetic expressions.

Examples:

1) Print the sum and average of sales for quarters 1-3.

```
PRInt SUM(SALES(Q1-3)), AVE(SALES(Q1-3));

SUM = 200000.00

AVE = 66666.67
```

2) Compute and print the ratio of the maximum value of accounts receivable to the minimum value of accounts receivable over quarters 2-6.

```
COM A = MAX(AR(Q2-6))/MIN(AR(Q2-6)); PRI A;

A = 1.78
```

3) Print the natural log of temporary variable R (R's valule must be strictly greater than zero).

```
PRI LOG(R);
```

3.2.8 Using Batch Files (BATch, TERminal, REWind)

By using a technique unique to your computer system (but available in some form on virtually all systems), you may set up a file of SLIM commands and instruct SLIM to read its commands from this file. This will allow you to set up standard or "canned" retrievals that you wish to run each quarter. Your instructor will tell you how to set up the file.

Once the file has been created, the commands BATch, TERminal and REWind give you interactive control between the canned routine and the terminal. You can have SLIM execute the file by logging in and typing the command BAT;. SLIM will then read the file and execute commands, printing results at your terminal (except when a SUBset command is used). The TERminal command instructs SLIM to again accept input from the terminal rather than the batch file. The REWind command is used to reposition the batch file to the beginning so that it may be run again, perhaps with modified data.

Suppose, for example that every quarter you wanted to LISt sales volume, market share, price, and advertising for your firm. You could create a file containing:

```
LIS  SV,MK(P1-2,A1-2),PRICE(P1-2,A1-2),ADV(P1-2,A1-2);TER;
```

Then, once you had logged in, you could type BAT; and SLIM would read and execute this file. When SLIM is finished, control will return to the terminal.

3.2.9 REMark

The REMark command is used to augment standard SLIM output labeling. It is used primarily with batch files to assist you in remembering the definition of temporary variables or other items output from a canned file. The general form of the command is:

```
REMark < user remarks >
```

When this command is encountered SLIM simply prints the remarks at your terminal. This is the only SLIM command that does <u>not</u> end with a semicolon. There must be at least one blank between REMark and the remark itself. Preceeding and embedded blanks are retained in the output to allow some formatting. For example by including the proper number of blanks you can prepare headings or footings for LIST commands.

It is recommended that the batch file always end with the following sequence of statements:

```
TER;
REM SLIM HAS COME TO THE END OF THE TRAIL
REM THE BATCH FILE IS FINISHED
TER;
REWIND;
```

This sequence of statements will protect you from inadvertently attempting to read beyond the end of the batch file. Suppose, for example, that you execute the batch file in the paragraph above and then attempt to execute it again, but forget to REWind it. On most systems you job will abort (SLIM will die) and you will have to login again.

If, however, you include the recommended commands, on the second attempt SLIM would print the REMarks:

```
SLIM HAS COME TO THE END OF THE TRAIL
THE BATCH FILE IS FINISHED
```

and return control to the terminal. If you still forget to REWind the file, the final command (REW;) will cause SLIM to REWind and then execute the file.

The use of canned queries and a simple model is illustrated in Figure 3.1. In this example the user wants to know how much cash will be available next quarter under various assumptions of sales volume and price. In BML, cash is available from three sources: revenue from current sales and backorders for each of two products in two market areas (approximately 60% collected in cash), accounts receivable (100% collected) and cash-on-hand. Backorders are filled at the lower of current price or price at the time of order.

Figure 3.1
Sample SLIM Dialog
(User typed entries are underscored)

```
HOWDY, I'M CALLED SLIM.
WHAT IS YOUR FIRM NUMBER?
2
PASSWORD?
XXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXX
 ...
BATCH;
 ...
 Y IS PRICE CHANGE AND Z IS SALES VOL CHANGE
 ...
 X IS SALES REVENUE
 ...

      PRIC          Y           SV          Z           X
     F2,Q8                     F2,Q8

     42.50       0.0000      9070.00      0.0000    385475.00
     11.00       0.0000      6833.00      0.0000     75163.00
     44.00       0.0000      5547.00      0.0000    344068.00
     12.00       0.0000      2254.00      0.0000     27048.00
 ...
 R IS CASH FROM SALES, SUM IS TOTAL CASH AVAILABLE
 ...
R = 439052.40
 ...
AR(F2,Q8) = 291151.64
 ...
CASH(F2,Q8) = 46931.90
 ...
SUM = 777135.95
 ...
COMPUTE Y(1)=2.0; COM Z(1)=-1000;REWIND;BATCH;
 ...
 Y IS PRICE CHANGE AND Z IS SALES VOL CHANGE
 ...
 X IS SALES REVENUE
 ...
```

```
     PRIC          Y          SV          Z          X
     F2,Q8                    F2,Q8

     42.50      2.0000     9070.00   -1000.00   359115.00
     11.00      0.0000     6833.00       0.0000  75163.00
     44.00      0.0000     5547.00       0.0000 244068.00
     12.00      0.0000     2254.00       0.0000  27048.00

 ...
  R IS CASH FROM SALES, SUM IS TOTAL CASH AVAILABLE
 ...
 R = 423236.40
 ...
 AR(F2,Q8) = 291151.64
 ...
 CASH(F2,Q8) = 46931.90
 ...
 SUM = 761319.95
 ...
END;
 SLIM SAYS SO LONG, PODNER.
```

The command file is listed in Figure 3.2. The file has been set up so that the user can interactively define variables Y(1) through Y(4) to be changes in price, with variables Z(1) through Z(4) to be changes in sales volume. The manner in which this is done is explained later. The first four lines of the command file define variables X(1) through X(4) to be the new trial prices (current prices plus the changes). Trial sales volumes (X(5) through X(8)) are calculated on lines 5 through 8. Finally, revenue from sales and backorders is computed on lines 9-12. Notice how the MIN function is used to select the appropriate price on backorders.

Figure 3.2
SAMPLE SLIM BATCH FILE

```
COM X(1) = PRICE(P1,A1) + Y(1);
COM X(2) = PRICE(P2,A1) + Y(2);
COM X(3) = PRICE(P1,A2) + Y(3);
COM X(4) = PRICE(P2,A2) + Y(4);
COM X(5) = SV(P1,A1) + Z(1);
COM X(6) = SV(P2,A1) + Z(2);
COM X(7) = SV(P1,A2) + Z(3);
COM X(8) = SV(P2,A2) + Z(4);
COM X(9) = X(1) * X(5) + BO(P1,A1) * MIN( PRICE(P1,A1),X(1) );
COM X(10)= X(2) * X(6) + BO(P2,A1) * MIN( PRICE(P2,A1),X(2) );
COM X(11)= X(3) * X(7) + BO(P1,A2) * MIN( PRICE(P1,A2),X(3) );
COM X(12)= X(4) * X(8) + BO(P2,A2) * MIN( PRICE(P2,A2),X(4) );
REMARK Y IS PRICE CHANGE AND Z IS SALES VOL CHANGE
REMARK X IS SALES REVENUE
LIST PRICE(A1-2,P1-2),Y(1-4),SV(A1-2,P1-2),Z(1-4),X(9-12);
REMARK R IS CASH FROM SALES, SUM IS TOTAL CASH AVAILABLE
COM R = SUM(X(9-12)) * .60; PRI R,AR,CASH,SUM(R,AR,CASH);
TER;
REM SLIM HAS COME TO THE END OF THE TRAIL
REM THE BATCH FILE IS FINISHED
TER;
REWIND;
```

The REMark command is then used to augment the standard headings of the LISt command which is used to output results. Finally, total sales revenue, cash from sales revenue, accounts receivable and cash-on-hand are printed at line 17 and control returns to the terminal at line 18.

In the sample terminal session in Figure 3.1, the user first assumes no change in price or sales volume. SLIM automatically sets all variables to zero initially, so after logging in the user immediately types in BATch; and SLIM begins executing commands from the user-prepared batch file. The first output the user gets corresponds to the assumption of no change in price or sales volume. The user can then play "what if..." games by redefining price changes and/or sales volume changes using the COMpute command. The user must then REWind the file and enter BATch again. In the example, the user raises the price on product 1 in area 1 by $2.00 and lowers sales volume 1000 units, then reruns the command file and logs off.

3.2.10 WHEn

The WHEn command is used to perform SLIM operations contingent upon the comparative value of two data base items, temporary variables, functions, or constants. The general form of the command is:

```
WHE <item> <operator> <item> ; <then statement> ;
<else statement> ;
```

Where item is a single-valued (i.e. no range on subscripts except in functions) data base item, temporary variable, function or constant, and the operator is a logical operator selected from Table 3.3.

Table 3.3
Logical Operators

Operator	Meaning
<	Less than
>	Greater than
=	Equal
=>or>=	Greater than or equal to
=<or<=	Less than or equal to
<>or><	Not equal

The "then statement" is any (one) SLIM command that will be executed if the result of the WHEn condition is true and will be skipped if the WHEn condition is false. The "else statement" is also any SLIM command. It will be executed regardless of the WHEn condition result.

Examples:

1 If firm 4 had positive earnings in quarter 6, print taxes in quarter 7, then print earnings in quarter 7; otherwise, only print earnings in quarter 7.

```
WHEN BL(F4,Q6)>0; PRI TAX(F4,Q7);
```

```
PRI BL(F4,Q7);
```

If firm 4 had earnings of $10,000 in quarter 6, taxes of $3,000 in quarter 7, and earnings of $5,000 in quarter 7, the output would be:

```
TAX(F4,Q7) = 3000.00
BL(F4,Q7) = 5000.00
```

On the other hand, if earnings in quarter 6 were -$8,000 (a loss), output would be:

```
BL(F4,Q7) = 5000.00
```

2 If temporary variable A is equal to 1, accept input from the batch file and then add 1 to A; otherwise, don't go to the batch file, but simply add 1 to A.

```
WHEN A=1; BATCH;
COM A=A+1;
```

3 For quarter 8, if the productivity (sales revenue per sales rep) of my firm's sales reps is less than the industry average, print commissions paid by all firms; otherwise, exit from SLIM.

```
REM COMPUTE MY SALES REPS PRODUCTIVITY
COM P=SALES(Q8)/SUM(SREP(Q8,A1-2));
REM COMPUTE INDUSTRY AVERAGE
COM A=AVE(SALES(Q8,F1-*)/SUM(SREP(F1-*,A1-2,Q8));
WHEN P<A; PRI COMM(P1-2,F1-*);
END;
```

3.2.11 Indirect Subscripting (F+ and Q+)

An option called indirect subscripting is available to provide variable subscripts for firm and quarter. This option is particularly useful with batch files as explained subsequently. To use indirect subscripts, the COMpute statement is used to

define a value for temporary variable F or Q. When SLIM encounters a + after an F in a subscript, it then uses the value currently stored in temporary variable F for the firm subscript value. Likewise, when Q+ is encountered, SLIM uses the current value of Q to define the quarter for which data is desired.

Examples:

1 Use indirect subscripting to print the BullBear Index for quarter 4.

```
COM Q=4;
PRI BBI(Q+);
```

2 Use indirect subscripting to print accounts receivable for firms 3 and 6, in quarters 7 and 8.

```
COM F=3; COM Q=7;
PRI AR(F+,Q+);
COM Q=Q+1;
PRI AR(F+,Q+);
COM Q=Q-1;
COM F=6;
PRI AR(F+,Q+);
COM Q=Q+1;
PRI AR(F+,Q+);
```

The principal use of indirect subscripting is with batch files. For example, with indirect subscripting and the REWind command you can create simple programs (a set of commands that are executed several times). You can also set up canned programs which allow you to compare output to the results in the same quarter a year ago.

Examples:

1 Set up a batch file which will determine which firms had losses in the current quarter, and the amount of the loss.

```
REM THIS IS A BATCH FILE
COM F=F+1;
WHEN F>5; TER;
WHEN BL(Q*,F+)<0; PRINT BL(Q*,F+);
REWIND;
```

This assumes there are 5 firms. If firms 2 and 4 each had losses of $5,000, output would be:

```
BL(Q12,F2) = -5000.00
BL(Q12,F4) = -5000.00
```

2 List sales reps in area 1 and 2 for the current quarter (12) and for the same quarter a year ago.

```
REM THE NEXT COMMAND COMPUTES THE SEQUENTIAL
REM QUARTER NUMBER OF THE CURRENT QUARTER
COM Q=(YEAR-YEAR(Q0))*4+QTR-QTR(Q0);
REM SUBTRACT 4 TO GET SAME QTR 1 YEAR AGO
COM Q = Q-4;
LIST SREP(Q+,A1-2),SREP(Q*,A1-2);
TER;
```

3 Determine which firm (of 6) had the minimum price for product 1, area 1, in the current quarter (12):

```
COM A=MIN(PRICE(F1-6));
COM F=F+1;
WHEN F>6; TER;
WHEN PRICE(F+)=A; PRINT PRICE(F+);
REW;
REM SLIM HAS COME TO THE END OF THE TRAIL
REM THE BATCH FILE IS FINISHED
REW;
```

This is somewhat inefficient since SLIM will actually execute the first line 7 times when it really needs to be executed only once. To avoid this, you could set A to the minimum price interactively before calling the batch file. The first line could then be omitted from the file.

4. Optimizing the Use of SLIM

While SLIM will permit considerable variety in the commands you may give, some prior planning may produce more satisfaction than merely making queries at random. These suggestions are all optional, but most users will be happier with them than without them.

4.1 Sequencing Subscripts

As indicated earlier, subscripts may be given in any order. However, the order may influence the response time of the query for many computer systems. The data base is sequenced by quarters, and SLIM will fill its data tables with one quarter's information each time it reads from the data base file. As a consequence, if SLIM can gather all desired information from a given quarter prior to moving on to read the data for the next quarter, execution will be more rapid. On the development system the command PRI PRICE(Q1-*,F1-*) has a much better response time than the command PRI PRICE(F1-*,Q1-*). The first example reads the price (assumed P1, A1) for each firm in Q1, then moves on to read the price for each firm in Q2, etc. The second example reads the price for firm 1 in Q1, then goes on to read the price for firm 1 in Q2, etc. After reading all periods to obtain the price for firm 1, it repeats the process for each additional firm. Thus, if there are 6 firms, the second command reads through the data base 6 times, whereas the first command reads through the data base only once. This difference can be very significant on many systems.

4.2 Putting Labels on Output

It is very easy to quickly accumulate many different reports from SLIM, and then to become confused as to which report is from which period etc. It is a good idea to begin each session by PRInting the current QTR and YEAR values at the top of the report. You could also LISt or SUBset QTR and YEAR as a means of labeling each listed or subset outpuut, as appropriate. Because QTR and YEAR do not provide for subscripts, they must be LISted as a separate command without subscripts, and only the current QTR and

YEAR may be so LISted. They would also need to be placed in a separate SUBset command.

APPENDIX A

Error Codes

1. Illegal command
2. Bad arithmetic expression, an integer constant not less than 1 nor greater than 16.
5. A constant may not have more than one decimal point
6. IFLAG is out of range
7. A constant is limited to 14 non-decimal places.
8. A constant must be followed by either a space or a symbol.
9. An unrecognized symbol has been encountered.
10. Each command has a limit of 10 constants; more than 10 have been encountered.
11. Left side of COMpute is not a temporary variable.
12. The '=' is not present in a COMpute.
13. No ';' at the end of a line.
14. Illegal sequence of operators and operands in a compute.
15. Bad index.
16. Unmatched parentheses in temporary variable.
17. Illegal argument type.
18. Duplicate argument type.
19. No ',' or ')' after argument.
20. No 'J' position argument allowed.
21. Illegal 'I' positon argument.
22. Illegal name.
23. Symbol table too large.
24. Illegal firm or quarter.
25. Delay not elapsed for data base item.
26. Illegal function argument.
27. Illegal argument range.
28. Argument beyond limits.
29. Unimplemented command key word.
30. Command too long.
501. Too many columns in a SUBSET command.
601. Not all columns equal length in a LISt or SUBSET command.
602. Attempt to LISt a function.
603. Too many columns in a LISt command.
625. Attempt to print a constant.
650. Illegal function code.
700. Unintelligible WHEn command (not <>= combination?)

```
**************************************
*                                    *
*              S L I M               *
*                                    *
*                                    *
*       System Laboratory for        *
*       Information Management       *
*                                    *
*  Business Management Laboratory    *
*       Data Management System       *
*                                    *
**************************************
```

****

Administrator's Manual

****

by

James F. Courtney, Jr.
Texas Tech University

and

Ronald L. Jensen
Emory University

Preliminary Version
Rev. August 1979

PREFACE

In this initial version of SLIM we have attempted to include the essential elements of the query portion of a decision support system. We view it as the beginning of representative software for managerial decision support rather than the final product. We expect that SLIM will continue to evolve, and would welcome correspondence from users with your views on how the entire package may be improved. We ask that you remember that SLIM was designed to be transportable, which requires that language features etc. be of the least common denominator variety. We know that there are more efficient ways to program parts of SLIM, but the present code was chosen to avoid any features which might not be available on all systems.

Development of the SLIM program was the work of several persons, to whom proper recognition needs to be given. In particular, the contributions of Frank Vogler, Gordon Rapkin, Ginger Breedlove, and David Paradice need to be acknowledged. While SLIM received the benefit of input from several persons, the authors accept the responsibility for the errors or ommissions.

[The SLIM program is still considered to be in a testing phase, and while the authors accept responsibility for correcting errors, no warranty is made as to its accuracy or completeness.]

Ronald L. Jensen
Atlanta, GA

James F. Courtney
Lubbock, TX

Rev. August 1979

Table of Contents

Systems Laboratory For Information Management
Administrator's Manual

by
James F. Courtney
Texas Tech University

Ronald L. Jensen
Emory University

1. Introduction

Our approach to teaching management information systems begins with use of a business simulation known as the Business Management Laboratory (BML). BML is used to establish a managerial decision-making context for the course. The simulation is complex enough to make the students feel as if they are in a semi-structured environment. This environment motivates the need for decision-making aids in a manner that is meaningful and relevant to the students.

In the first two to three weeks of the course some concepts of management information systems and decision support systems are introduced. Then the students are introduced to the really unique aspect of the approach: a software package that can be used to develop an MIS/DSS. This package is called SLIM (the Systems Laboratory for Information Management). SLIM is a model of a commercial data base management system. It consists of an integrated data base that is built as the game is played, a conversational query language used to access the data base and create simple reports, and software for administering the process. The SLIM query language is simple and has been mastered by nontechnical students in a few hours. No prior exposure to computers or programming is required.

Compared to commercial packages, the BML/SLIM system is small and inexpensive. BML/SLIM will execute on very modestly sized minicomputers (or large machines) and can be obtained for essentially the cost of reproduction. Commercial systems generally run on large machines and cost from $25,000 to $100,000.

It should emphasized that SLIM is <u>not</u> a set of canned report generator programs. Rather than canned reports, it provides a

data base and a generalized data access and data analysis capability. Students must use SLIM to interactively query the data base and to create reports to be used in decision-making procedures for BML. That is, students develop their own decision procedures then use SLIM to develop decision support systems.

BML was first published several years ago and has been widely used (currently at about 100 schools). About three man-years of effort went into the development of SLIM which has only recently been released. The programs are written in ANSI FORTRAN IV and have been designed to be highly portable.

1.1 Benefits of the BML/SLIM Approach

There are many significant benefits provided by the simulation approach. Among these are:

1. Students get "hands-on" experience in the development of management information and decision support systems using a software package (SLIM) similar to those currently in commercial use.

2. Students use the MIS/DSS they develop in a lively and realistic context provided by BML. BML also serves to integrate material from virtually the entire business curriculum.

3. The SLIM query language is simple but powerful and can be mastered by nontechnical students in a few hours. (The language can be covered in class in 2-3 hours).

4. Course evaluations indicate that students find BML to be very enjoyable and challenging (much preferable to ordinary lectures and cases).

5. When faculty resources are limited, BML/SLIM can be used in policy courses to expose students to MIS material that might not otherwise be available. Furthermore, the system can be used to emphasize the interaction of policy and strategy and the MIS/DSS.

6. By varying the emphasis, BML/SLIM can be used in a wide variety of classes ranging from policy to data base management.

7. The portability and small size of the programs makes it possible for schools with limited computer facilities to use the system at little expense.

8. In more technical courses the programming techniques in BML/SLIM can be used as classroom examples. Also, the data base and programs can be used as a basis for term projects.

This manual describes the administrative procedures for classroom use of the SLIM modified version of the Business Management Laboratory (BML). The modified version of BML will be denoted BMLSM in this manual (for BML-SLIM Version). It is assumed in the manual that the reader is familiar with BML administrative procedures (as described in the BML Administrator's Manual, Business Publications, Inc., Revised Ed.). It will probably be convenient to have read the SLIM User Manual also.

The major modification to BML is the addition of a subroutine that creates a data base as iterations of the game progress. SLIM is a stand-alone program (separate from BML) which processes queries entered in a user-oriented language and extracts the requested information from the data base.

Since a good deal of latitude is available regarding the amount and depth of technical material covered in class, SLIM can be used in courses for non-MIS majors (such as introductory EDP or policy) as well as in MIS, DSS, or data base courses. Some suggestions for classroom use of SLIM are discussed in section 4.

1.2 Structural Information Flow

The general flow of information in this system is illustrated in Figure 1.1. As shown, students make their decisions for a round of play and the decisions are input to the revised BML program. BMLSM processes the student input, creates standard financial reports, the new history file and updates the data base to be used by SLIM. Students can then use the SLIM query language to access the data base.

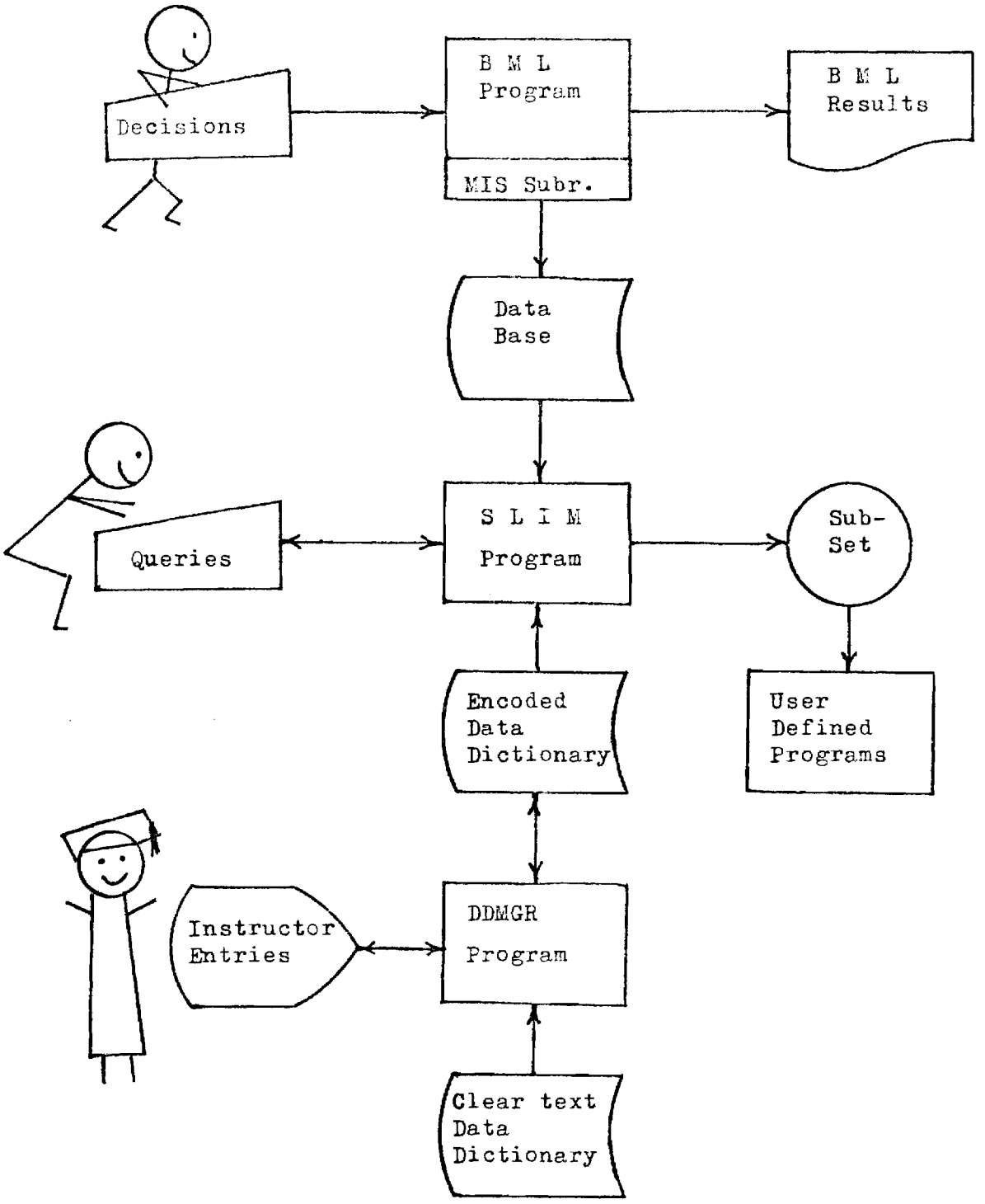


Figure 1.1

BML/SLIM

Program Structure

1.3 Computer System Requirements

The computer system on which BMLSM and SLIM are to be run must satisfy the following requirements:

1. Have interactive access (access via terminal). Hard copy terminals work best.
2. Be capable of compiling ANSI FORTRAN IV programs
3. Have approximately 100k bytes of core storage.
4. Have approximately 600 bytes/firm/quarter of mass storage for data (preferably direct access but tape will work) and some additional storage for programs.

2. Set-Up Procedures

In order to run BMLSM and SLIM, the administrator must (1) make minor changes to the BML data decks (in addition to the procedures described in the BML Administrator's Manual); (2) make minor changes to the BML program deck (which is currently set-up to run BML without the interface with SLIM); (3) set device selection statements in the DDMGR program and (4) set the device selection statements in SLIM. Each of these actions is discussed in detail in this section. First, however, some instructions are given on how to convert existing BML programs to BMLSM. These change instructions are applicable only to those administrators who already have BML and wish to convert to BMLSM without acquiring the entire BMLSM program.

2.1 Converting BML to BML-SLIM Version

If you already have BML running on your computer system, you can convert to a version that is compatible with SLIM by making a few modifications and by acquiring and installing the SLIM Program and Data Dictionary. By far the most extensive modification is inclusion of SUBROUTINE MIS which creates the SLIM data base. A listing of this subroutine is given in Appendix A. Next, follow the instructions in Section 2.2 on setting up BMLSM data decks. Then, follow the instructions in Section 2.3 on setting up the BMLSM program deck, except that when activation of a statement is required, you will have to insert that statement in the appropriate place. Finally, follow the procedures in Section 2.4 and 2.5 on initializing the SLIM Data Dictionary and installing the SLIM program.

2.2 Setting-UP BMLSM Data Decks

There are only two differences between BML data decks and the data deck for BMLSM. These differences come into play only once--prior to the first BMLSM run. These differences are:

1. A 1 (or any legal non-zero value) must be placed in the fifth field of the BMLSM history deck header card, as shown in Figure 2.1. This is the second card in the history deck (the title card is the first). A non-zero value here indicates that no data base yet exists. After the first BMLSM run this value is automatically set to zero and will remain zero thereafter.

Figure 2.1

HISTORY DECK HEADER CARD

Cols.	Content	Format
1-1	ZRMC(1) The contract price for futures purchases of raw material 1.	F10.0
11-20	ZRMC(2) The contract price for futures futures purchases of raw material 2.	F10.0
21-30	ENDEX The economic index for this period.	F10.0
31-40	PRR The prime interest rate for this period.	F10.0
41-50	DISC Non-zero value indicates no SLIM data base yet exists	F10.0
51-60	dummy, not used	
61-70	dummy, not used	
71-80	the label HST HDR 1	

2. Two additional device numbers must be included on the device selection control card (the first card of the BMLSM parameter deck), as shown in Figure 2.2. One input device (corresponding to the "old" data base) and one output device (the "new" data base containing output from the latest run) have been added. SLIM is then used to access the new data base.

FIGURE 2.2

THE BUSINESS MANAGEMENT LABORATORY

PARAMETER CARDS

Card 1

DEVICE SELECTION CONTROL CARD

This card must precede all other cards in the data deck, and identifies the local system device numbers for the logical device selections used in the program.

Cols.	Data	Value	Format
1-5	LIN : The primary input devices, most likely the card reader. The remaining parameter values are read from this device.	5	I5
6-10	LI2 : Second input device, used to read the history file. May be the same as LIN.	5	I5
11-15	LI : Third input device, used to read the decision cards. May be the same as LIN.	5	I5
16-20	LI4 : Fourth input device, use for old SLIM data base	8	I5
21-25	LO1 : The output device for the program results, usually the highspeed printer.	3	I5
26-30	LO2 : The output device for the history "deck" used to link one quarter to the next. This is usually a card punch, as its output is read by the device LIN.	7	I5
31-35	LO3 : Third output device, used for new (updated) SLIM data base.	9	I5

After the initial set-up, the administrator no longer needs to concern himself with these data fields. The procedures for running BMLSM are exactly as those described in the BML Administrator's Manual. The history deck and the student input deck are exactly the same as for ordinary BML.

2.3 Setting-Up the BMLSM Program

The BMLSM program deck that you receive will actually be set-up to run pure BML (without the SLIM interface). The statements necessary to run BMLSM (the SLIM Version) are included in the deck (for those decks supplied with SLIM), but have been deactivated by the inclusion of a C in column 1 so that the compiler treats them as comments. These statements will have to be activated by removing the C. Also, certain statements will have to be deactivated by removing them from the deck or by placing a C in column 1. It should be noted (for your information only) that SUBROUTINE MIS has been added for creating and updating the SLIM data base. No C's were put in column 1 of this routine because it will not be called unless the call in BML (described below) is activated. The following steps must be taken by the administrator:

1. Activate: CALL MIS This statement is in the main routine after the CALL RNDP statement labeled BML0032 (labeling is in columns 72-80). This statement calls the subroutine which creates the SLIM data base.
2. COMMON LI4, LO3 has been added in every subroutine This statement has been added to BML to facilitate the use of SLIM. It is "benign" if SLIM is not used, but must be added as the last line in the COMMON declaration of each subroutine if you are converting an earlier copy of BML.
3. Activate: READ(LIN,12) LIN, LI2, LI3, LI4, LO1, LO2, LO3 This statement is in SUBROUTINE LOAD at line LOD0037-. It replaces a similar statement which did not include LI4 or LO3. This replacement must be made if an early version of BML is being converted.
4. Activate: READ(LI2,10)ZRMC(2),ENDEX,PRR,DISC This statement is in SUBROUTINE LOAD at line LOD0055*. It

replaces a similar line which did not contain the variable DISC. This line must be changed if an old version of BML is being modified. This statement reads the history deck header card. The last variable (DISC) is the flag indicating whether a data base exists (zero) or if this is the first run (non-zero). DISC is EQUIVALENCED to MISFL in SUBROUTINE MIS.

2.4 Setting-Up the DDMGR Program

An interactive program called DDMGR is used to initialize and manage SLIM user ID's and the data dictionary (both reside on the same file). This file should be carefully protected by the administrator since students could gain access to other firms' data if this file is compromised. A variety of measures is used to provide security. The use of DDMGR and the security measures are discussed in Section 3. Only the steps necessary to set-up DDMGR are discussed here (DDMGR must be run before SLIM can be used).

Device Assignments for DDMGR

Before DDMGR is compiled and executed, logical device codes must be set. Values for FORTRAN variables corresponding to the device codes for the administrator's computer system must be established. The FORTRAN statements which must be changed are at the very beginning of the DDMGR program following the comments:

```
C THESE ARE LOGICAL DEVICE ASSIGNMENTS
C THEY MUST BE SET ACCORDING TO THE
C ADMINISTRATOR'S COMPUTER SYSTEM BEFORE COMPILING
```

The variables names used are:

TRMIN - terminal input
TRMOUT - terminal output
DDIN - Data dictionary input from the card deck supplied with SLIM
DDCODE - Encoded data dictionary actually used by SLIM
LPRNT - Line printer, used for listing data dictionary during

initial load
DBIN - Uncoded SLIM data base
DBCODE - Coded SLIM data base (only used when encryption option is used-see section 3)

Once the device assigmments have been set, you are ready to compile and execute DDMGR. The first run will involve initialization of the data dictionary. See Section 3 for instructions.

2.5 Setting-Up to Compile SLIM The administrator must also set the six device selection statements in SLIM. Presently these are lines 27 to 33 of the main program. Variables and corresponding files are:

IBAT - File for use with BATCH Commands
TRMIN - Terminal input
TRMOUT - Termnal output
ISYMTB - Data Dictionary File
IDABSR - Data Base
IFILE - File for SUBSET opertions

The SLIM program includes write statements to blacken out the password at login time. These statements use the carriage control character "+" to inhibit line feeds. Some systems ignore carriage control characters sent to terminals. If this is the case on your system you may wish to remove these statements from the program. They are lines 2437 and 2439 of subroutine VERIFY (WRITE (TRMOUT,9000)).

Also included but deactivated by the use of a "C" in column 1 are statements useful in debugging. If you encounter problems, it may be helpful to recompile with the C replaced with a blank. The debug statements are presently directed toward absolute device number 2, rather than a variable device number (WRITE (2,???)). You may need to change this device number to fit your system.

2.6 Summary

To summarize, the administrator must perform the following tasks when setting-up BMLSM/SLIM.

1. Follow the set-up procedures for the BML parameter deck, history deck and decision deck with the two exceptions noted in Section 2.2 (i. e., the data base flag, and the two additional device assignments).
2. Activate/deactivate the appropriate statements in BMLSM (Section 2.3).
3. Set device assignments and compile DDMGR. (Section 2.4)
4. Set the device assignments and compile SLIM (Section 2.5).

3. SLIM Administrative Procedures

After setting-up the programs as described in Section 2, there are few additional administrative procedures that are necessary in running SLIM (other than those associated with running BML). The one thing that must be done is initialization (as described below) of the data dictionary and user ID's by using the DDMGR program. You will have to set the number of active user IDs to the number of firms + 1, and select user IDs for each team. This can be done either by using those IDs provided with the BMLSM/SLIM system, or by substituting IDs that you or your students choose. Substituting your own passwords can be done before the data dictionary is initialized or by using DDMGR to change passwords after initialization as described below.

Of course you will also have to instruct your students in the use of installation dependent operations. These include details on logging in on your computer, automatic system prompts such as ? or *, and setting-up and interfacing with SLIM batch files.

There are several options available to the administrator through DDMGR. These include:

1. The option to change user ID's at any point.
2. The option to encrypt the data base after each round of play, if additional security is desired.
3. The option to change the number of active user ID's.
4. The option to obtain a list of active user ID's.

The required initialization procedures and options are discussed below.

3.1 Initializing the Data Dictionary and Using DDMGR

After the device assignments have been set and the program compiled, DDMGR must be run to initialize the data dictionary. In the initialization process, DDMGR reads the clear text data dictionary file provided with the BML/SLIM system and writes a new, encoded file that is actually used by SLIM (and henceforth by DDMGR). During initialization, DDMGR performs the following tasks:

1. Encodes user ID's (by substituting one alphabetic character for another wherever it occurs) and scrambles

the sequence of encoded characters in the ID.
2. Sorts the data dictionary.
3. Computes check-sums for rows and columns of the data dictionary, so that SLIM can test to see if the data dictionary has been tampered with. If so, SLIM prints the message "The Data Dictionary has been modified," and stops. If this occurs the administrator must reinitialize the data dictionary using DDMGR.
4. Writes the new file in binary, so that it may not simply be listed, should a student gain access to it.

DDMGR requires the Administrator to enter a password to permit it to execute. This provides one level of protection against unauthorized use of the program.

The administrator should set the number of active users to the number of firms + 1 before the initialization run. The file always contains nine user IDs, but any number from 2 (one firm and the administrator) to 9 may actually be in use. Then DDMGR should be executed. The program firsts asks you to enter the administrator's number and password (00,ADMIN on the original data dictionary). You are then asked if this is the initial load (enter 1 for yes, 2 for no). If it is, the password you typed in is checked against that on the input deck. You get three attempts to get the user number and password correct. If this is not the initial run, the password is checked against the encoded data dictionary (rather than the input file).

On the initial load, a call is made to subroutine MSLOAD. MSLOAD reads the uncoded data dictionary, encodes and scrambles the user numbers, passwords, and user names and writes them (in binary) to the new data dictionary to be used by SLIM. Before the data base items and corresponding information is written, a call to SORT is made and the items are alphabetized using the collating sequence on your machine (this varies among machines and SLIM expects the dictionary to be sorted). Next STPUT is called. STPUT computes check-sums for each row and column of the dictionary (excluding user numbers and passwords) and writes the data item information on the encoded file. Also, the data dictionary is listed on the LPRNT file.

Control returns to DDMGR and you are asked if you want to:

1. Change a user number/password
2. List user numbers and passwords

3. Stop
4. Change number of active users
5. Encrypt the data base

If you enter 1, DDMGR will ask for the firm number whose user ID is to be changed (see Section 3.3 for the format of the user ID). Then, the new user number and password is solicited and the change is made. If a 2 is entered, current user numbers and passwords are listed (with any changes made).

When a 3 is entered, if any changes have been made, the encoded data dictionary is written-over with the new information; otherwise, DDMGR terminates. Notice that changes will not be reflected automatically in the original data dictionary supplied with SLIM. Any updates to passwords will only be incorporated on the encoded file.

If a 4 is entered, DDMGR will ask "HOW MANY FIRMS?" The Administration indicates how many active teams are to be using SLIM (1-8). This permits DDMGR to check to be certain that a change request is not for an inactive firm and to list only the active firms when the listing options (2) is used.

The Data Dictionary is described more completely in section 5.1.

It is strongly recommended that you do not keep a copy of this "clear text" data dictionary on the computer where students may have access to it. DDMGR is constructed to permit this file to be read once, then it is encoded, and further operations are carried out using the coded file. This file consists of four sections:

1. A header line to indicate the number of active users to be permitted. This is a maximum of 9, including the Administrator. This is followed by the 9 clear text passwords. The administrator needs to know the first of these to execute DDMGR.
2. The second section contains 7 numbers which identify various parameters of the data base. These are:

   -Number of data base elements (NSTEL)
   -Number of environmental factors(NENVF)
   -Number of firm factors (NFF)

-Number of firm factors in blocks (NFF1,NFF2,NFF3)
-Number of functions (NFUN)

3. The third section contains the data dictionary elements, including various access and delay elements.
4. The final section contains the letters of the alphabet, SLIM verbs, and special characters used by the program.

3.2 User ID's

User ID's are 36 characters long and consist of a user number, password and user name. The user number consists of two digits. The first is an industry number, primarily for convenience of the administrator when more than 8 teams are used, so that more than one "industry" must be run. The second digit is the firm number (1-8). When there is more than one industry, a data dictionary must be set up for each one.

The password consists of 9 characters and the user name is 25 characters. The characters may be letters of the alphabet, digits, blanks or commas.

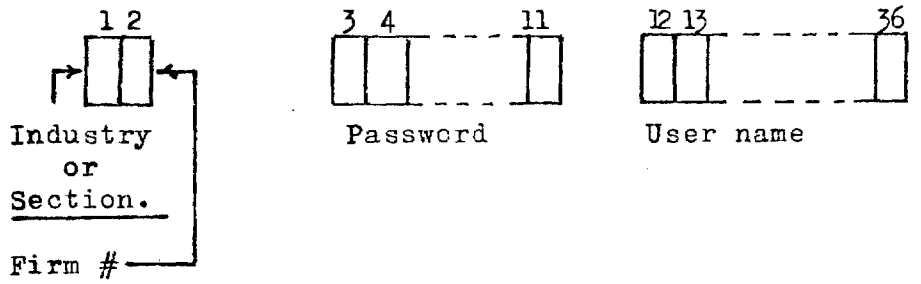


Figure 3.1
User ID Format

Only the user number and password (the first 11 characters) are used by SLIM to identify the user when he/she logs in. The user name is not entered at log-in time, but the characters in the user name are scrambled with the user number and password by DDMGR before the write to the encoded data dictionary is made. This provides an extra measure of security. Although the user name could be all blanks, it is suggested that this not be done, since this would be apparent to any student that gained access to the

data dictionary file and could be helpful in decoding passwords. It is recommended that the user name be used by the instructor to identify team members. For example, the decoded, unscrambled user ID for firm 5 in industry 1 might be

15BANKRUPT5 BOB, CAROL, TED, ALICE

The encoded, scrambled user id might appear as

QZRM9ATB5ALXZTONFE87CRWCVDIM321Z9SRN

when logging-in, the student would enter only

15BANKRUPT5.

Alternatively, the password could be the user number followed by an easily remembered phrase, such as

15,JOE,HAS,A,VERY,TIRED,HORSE

In this case the comma is part of the 9 characters used for the password and the remainder of the phrase (beyond ,JOE,HAS,) are used in the scrambling process. At login time the SLIM user would enter 15,JOE,HAS,.

A card deck is supplied with the SLIM system containing nine (9) user numbers and passwords and the data dictionary. There is a user ID for each of eight firms and one for the administrator. The administrator's ID and password in this deck is 00,ADMIN. The first two characters are zeroes.

The administrator may substitute other user ID's and passwords before the initial load of the data dictionary. If this is done, the sequence of cards for firms must be maintained (Admin, firm 1, firm 2, etc.). That is, the card for the administrator will always be first, firm 1 second, etc.

The administrator will always be firm 0 (zero). SLIM allows firm 0 to access any data without restriction. The administrator also has access to two data items not mentioned in the student manual. These are the BML cost reduction factor (CRF) and the R&D effectiveness index (RD). These are indexes internally computed by BML. They can be useful to the instructor in evaluating each firm's performance.

3.3 Encrypting the Data Base

Since student teams must have access to the data base, it is necessary to protect each firm's private data. The basic SLIM system provides only minimal protection by using unformatted FORTRAN write statements. It is tedious, but not impossible, for an industrious student to discover the structure of this file through a trial and error process, eventually gaining access to other firm's data. To prevent this, the administrator is given the option (through DDMGR) to have the data base encrypted. The process works as follows:

1. BMLSM is run in the standard mannner and a non-encrypted data base is created.
2. Any changes to user ID's are made by selecting DDMGR option 1.
3. The non-encrypted data base is read and an encrypted data base is created by selecting DDMGR option 5.
4. SLIM will recognize an encoding flag in the data base to reverse the process. Be careful to adjust any file commands used by your operating system so that SLIM will access the encoded data base.

The encryption process is sufficiently complex that it is highly unlikely to be broken by a student. Encryption is a function of 3 factors: each individual team's user ID (hence changes must be made prior to encryption), the sequential quarter number, and the position of the data item in the data base. Note that the entire data base is encrypted after each round of play. This is done in order to adjust for any changes to user ID's. The third field of the first record of the SLIM data base is a flag indicating whether the data base is encrypted. If this field is non-zero, SLIM assumes an encrypted data base.

The BMLSM program will only access the unencrypted data base file, and always sets the coding flag to zero. The DDMGR program must be used if this file is to be encrypted for student use. If option 5 is used, DDMGR always reads the unencrypted file and writes a new encrypted file.

4. Hints for Instructors

A good deal of latitude is available regarding the manner in which BML/SLIM is used in class. Perhaps the most common use is in MIS courses for business or management students who have already taken the core courses in the functional areas including some statistics, management science and computer programming. Use of BML/SLIM in such courses is discussed in some depth. Also, using BML/SLIM in policy courses and technical courses is briefly discussed.

4.1 Using BML/SLIM in MIS Classes

An outline for a paticular class is shown in Appendix E. The text referred to in the article was Burch and Strater's Information Systems: Theory and Practice (Hamilton, 1974). Any equivalent introductory text could be substituted.

The course begins with a brief introduction to systems theory in order to establish a conceptual framework for the discussion of MIS and DSS design. In this approach, divisions of an organization are treated as subsystems and the MIS is the main element linking these subsystems into an integrated whole. MIS systems analysis and design is discussed in this context. The DSS material is introduced exclusively through lectures and through the use of SLIM.

While the background material is being introduced, teams are formed and BML is started. SLIM is not used at this point, so the students must rely on manual procedures for decision-making. This serves to reinforce the need for automated decision-making aids that SLIM can provide.

Then the SLIM query language is introduced. It can be covered in two or three hours of lecture and even nontechnical students seem to easily master it. After the students have used SLIM for a while the course takes a more technical slant. Sequential, indexed-sequential, and direct file organization techniques are discussed. Here SLIM comes in handy to the instructor because the data base can be used as an example. For maximum portability sequential files have been used in SLIM; however, if the instructor's host computer system allows the use of ISAM or direct access files different access techniques can be discussed and possibly implemented by the class.

Next data structures are discussed. Again SLIM can be used as an example. The SLIM data base is a balanced tree. This structure can be compared and contrasted to other types available.

Introducing the SLIM query language before covering the technical material is beneficial to the instructor because it sparks the inquisitive nature of many students. That is, after the students have used SLIM, they become curious as to how it works. This is a great help in making the discussion of file organization techniques and data structures more meaningful.

The course concludes with a discussin of a complete commercial DBMS such as System 2000 or ADABASE and with presentations by each team. The presentations are designed to highlight the DSS that the team developed and its use in decision-making. In this course much more empahsis is put on the DSS than on the results of playing BML.

On the outline shown, the class played the game continously after it was begun. A useful modification would be to suspend play for one or two weeks after the students had become familiar with SLIM to give them an opportunity to work on the DSS.

Grading is based on two examinations, the team presentation, and a term paper prepard by each team. Like the presentation, the paper is usually 40% of the course grade. Since much of the work is small group oriented, each student is asked to evaluate the performance of every other student on the team. These evaluations are very useful in determining the final course grade.

Note that the results of playing BML are not used in the grading system for the course. This is done intentionally to direct emphasis towards the DSS as opposed to game play. Of course, it can be argued (and hopefully is true) that a good MIS should result in successful game play. However, if emphasis is put on game results, students may devote too much time to decision-making and not enough time to MIS development. The instructor may well be surprised at how competitive the students are about BML even when it is not a factor in the grade.

4.2 Using BML/SLIM in Policy Courses

Simulations have long been used in many schools in policy courses or other capstone courses to integrate material from various functional areas and disciplines. BML itself has been used at over 90 schools. BML/SLIM offers an efficient way to integrate

some MIS material into the curriculum when an entire MIS course cannot be offered.

In policy courses the major emphasis would be on BML results as opposed to the development of a DSS/MIS. However SLIM can be readily used to illustrate many MIS concepts and can also be a useful tool to the students in playing BML. Students in a capstone course in which an early version of SLIM was used were asked to rate its usefulness on a scale of 1 to 10. The mean rating was 7. There were indications that those who used it a lot rated it high and those who used it little or not at all rated it low. One of the lowest ratings came from a team which did not even pick up their password. No cause/effect measurement was performed at that time. Examinations (nongraded) indicated that the students learned a good deal about MIS/DSS even though very little class time was devoted to this material.

In simulations students must establish the organizational structure of their firms, allocate responsibility and authority, and, more importantly for our purposes, establish corporate objectives and policy and develop procedures for making the required decisions to implement such policy. We assume that a major objective of the course is to integrate functional areas and to relate the decision making process to the policy making function. The use of a DSS in conjunction with the simulation program can provide two benefits: (a) the mechanics of the decision making process are simplified so that the student may concentrate more on the policy and/or objectives aspects of the managerial decision process, and (b) the student begins to become accustomed to the concepts and ideas involved in establishing and using a DSS within the operations of a business. Some of the points relating to the MIS/DSS material which may be effectively brought out in the policy course could be:

1. The interaction of corporate goals and objectives with information used in the decision making process may be examined. In planning the MIS/DSS and in using it, the student becomes more accutely aware of the role which information plays in these important aspects of the managerial task. For example, the firm may set as an objective the achievement of some specified sales level by the end of the third period. The DSS may be used to examine the impact of various methods which they might use to achieve this goal, and time may be spent on considering the policy implications of these actions rather then just on the mechanical collection and processing of the data.

2. The firm may use the DSS to aid in selection of appropriate policies for their firm. For example, the DSS can be used to ask "what if" type questions for the purpose of evaluating the potential impact of alternative policy positions. This would be in addition to the usual philosophical questions which must be raised in considering policy questions. Where trade-offs are necessary between desires and possible achievement, this ability to explore alternatives can be of considerable importance to the manager.

3. The DSS may be used as an aid in assembling the data necessary to use some optimization tools, or perhaps in learning where such tools are appropriate in the decision making process. This can have important policy considerations in helping the firm to establish policies on matters such as where and under what conditions they would use satisficing rather than optimizing decision techniques.

4. The use of the simulation-DSS package gives students the opportunity to develop their own MIS/DSS within a corporate-like environment. In this sense it offers many of the same advantages quoted for the business simulation itself in that it provides a non-threatening atmosphere for learning as well as providing the experience of actually using the results of their efforts, providing the feedback which is so necessary in the educational process. This experience highlights the role of management in the development process and provides a good background for development of actual systems in the future.

4.3 Using BML/SLIM in Technical Courses

BML/SLIM can be an effective tool for introducing computer science and technical management students to the concepts of data base management. In such courses game play would be heavily de-emphasized and might only be done for seven or eight rounds of play to introduce the basic notions of query languages, data bases, and information requirements. The resultant data base and the BML/SLIM programs themselves can then be used as laboratories for the students' use in conducting experiments or doing term

projects. Some example term projects (rated as easy, moderate, or difficult) are:

1. Replace the sequential access feature in SLIM with random access or ISAM (moderate).

2. Use a commercial data base package (if available) on the SLIM data base. This might involve rewriting subroutine MIS in BML (moderate).

3. Apply the concepts of relational data bases to the SLIM data base (moderate).

4. Try to break security on the SLIM data base (using encryption) and/or the data dictionary (hopefully difficult).

5. Add one or more functions such as the standard deviation (moderate).

6. Build options in DDMGR for conversationally adding items to the data dictionary or changing information (delays, for example) already in the dictionary (moderate).

7. Develop interfaces to other software systems such as packages for statistical analysis, math programming, or financial planning (moderate to difficult).

8. Design a CODASYL-type data manipulation language for SLIM perhaps by using a pre-compiler (moderate to difficult).

9. Add new verbs to SLIM. For example, a BACKSPACE command to be used on batch files (easy) or CHANGE command to update the data base (difficult).

10. Write a program with appropriate editing to conversationally input decisions (moderate).

5. Technical Documentation

This section of the manual briefly describes how SLIM processes queries. The documentation is primarily for your information only, but will be helpful in debugging, should that be necessary. Only the general system structure and program flow is discussed. The source program is heavily commented and once the general structure is understood, you should be able to follow any subroutine by reading the comments.

5.1 Data Dictionary

Section 3.1 described the initialization of the Data Dictionary by the DDMGR program using the Data Dictionary supplied with SLIM. This section describes the structure of the Data Dictionary and what is involved in changing it.

The Data Dictionary file has four major sections. These are:

1. A header line followed by a set of nine password lines.
2. A set of parameters to indicate certain key values and the blocking of values within the data base records. These are explained below.
3. The set of data item (field) names used to access values within the data base and codes to identify certain attributes of these values.
4. Certain constant words and characters used by SLIM to interpret commands.

The section one header line indicates how many of the passwords are currently active. The remaining nine lines contain the original password entries: the administrator's followed by eight firm passwords for the clear-text file. In the encoded file used by SLIM the order of these passwords is changed in addition to encoding the passwords and user numbers. The Administrator should always be assigned user number 00. Individual firms are assigned two digit firm numbers. The first digit is an industry or section (in the sense of section of a course) number and the second is the firm number within a section (e. g., 11 refers to section 1 firm 1, 35 refers to section 3 firm 5).

The lines in section two indicate how many values are to be found in each of the records and/or key segments of the Data Dictionary and data base files. In the original package these

are:

1. The number of user elements (lines) in the symbol table
2. The number of fields in the environmental record for a quarter
3. The total number of fields for each firm in the quarterly records, divided into three physical records.
4. The number of items for a firm in the first physical record. These comprise 37 administrative and financial items for the firm.
5. The number of fields for a firm in the second physical record. These comprise 33 marketing items for the firm.
6. The number of items for a firm in the third physical record. These comprise 57 production and inventory related items.
7. The number of functions currently defined for SLIM. At this writing there are five functions defined (SUM, AVE, MAX, MIN, LOG).

The third section contains the alphanumeric symbols used by SLIM to locate each item in the data base or to identify a function operation. Each line in this section takes the form:

1. The first field contains a four character (or less) alphanumeric name. In a few cases more than four characters are given, but only the first four are actually used. These symbols are supported in reverse alphabetic order, which is the order expecteed by SLIM in searching for a name. A sort routine in DDMGR verifies that the correct internal collating sequence is maintained. Appendix D shows the table as supplied and as it is sorted in the order of item appearance in the data base. Note that the cost reduction factor (CRF) and R&D index (RD) are shown in these listings but not in the student manual.
2. The second field of the clear text data dictionary contains the variable name used for this item within the BML program. This field is not included in the encoded file, but is stripped by the DDMGR program.
3. The third field contains a numeric value to indicate the delay required before a firm can access this item from the record of any other firm. A zero indicates no delay, a 1 indicates a one period delay. A 99 indicates a 99 period delay, which is effectively a permanent block. Firm zero (The Administrator) ignores these delays, and

may access all data at all times.

4. The fourth field contains a numeric value which established the amount of "measurement error" to be built into this data when reported to any firm except the owner firm or the administrator. This should be viewed as the standard deviation of the error component, as the actual value may vary for each item. Repeated requests for a given data item will always return the same value.
5. The fifth field contains a numeric value which indicates the position of this item within the set of records for an individual firm. Thus SALES is the first data value in the first record of an individual firm's data, while STII (short term investment income) is the 25th item of data for that firm (it falls in the first physical record for that firm). If any changes are made in the data base (e. g., adding or deleting variables) these numbers must be adjusted as necessary to maintain the correct position pointers. A negative value or position indicates a SLIM function rather than a data base item.
6. The sixth and seventh columns contain a pair of numeric values to indicate the possible subscript range for this variable (I,J). These are called "ARRAY LIMITS." Thus a 22 indicates a 2 by 2 table, a 20 indicates a two element vector, and 00 indicates a single value (scalar). Note that a zero or a one would have the same meaning in this instance, as a zero length value is not possible by definition in SLIM.
7. The eighth column contains a numeric value to indicate the proper subscript type (material, stage, product) as a check when subscripts are specified. Firm number and quarter number subscripts are possible for all firm data, but other subscripts are checked for validity. Any item with a zero "IPOS TYPE" indicator may be reported only by quarter and firm (where appropriate). Note that these items also have zeros to indicate subscript range. If the subscript range is non-zero, this final value must be non-zero to indicate the type of subscript permitted. A 3 indicates it may vary by area (only), a 4 indicates it may vary by product (and by area, if the second array limit is greater than one). A 5 indicates that it may vary by material type (and by area, if the second array limit is greater than one). A 6 indicates that it may vary by production stage (and by area, if the second array limit is greater than one).

The fourth section contains a long vector of alphnumeric characters. These contain the key command words (END,COM,LIS,etc.), the full alphabet, the special characters used for operations or punctuation (*/+-, = etc.), and the ten numeric digits. These are stored in tables in SLIM so that the local computer's internal representation of a given character may be assured. The number of items in this vector and their sequence is extremely important, as they are "counted off" and directed to various internal arrays. Any change in this vector would require changing the loading routines, and possibly the size of one or more arrays in the COMMON area of all SLIM subroutines.

Should you wish to change any item except a current password, the change must be made in the clear text data dictionary, and the DDMGR program used to created a new operating data dictionary. Thus, for example, if a name is changed for a data base item, or delay is changed, or an error term is changed, DDMGR must adjust the various check-sums so that SLIM will accept the revised data dictionary. Note that the passwords in this revised operational data dictionary will revert back to the passwords given in the clear text file, so any variations (updates) in passwords must be entered using DDMGR, or the clear text file must be modified to correspond to the new passwords. Most users will find no need to change anything except the passwords, so the clear text file may be used once to create the operational (encoded) file, and DDMGR may be used thereafter to update the passwords. You may wish to put the clear text file in a secure (off line) storage after the encoded file has been created.

5.2 Data Base Structure

Logically, the SLIM data base has a hierarchic or tree structure, as illustrated in Figure 5.1. This is a balanced tree in that each parent node at a given level has the same number of children.

The root node is actually one three-value "header" record consisting of the number of quarters of data (ICQ), the number of firms (NF), and a code indicating whether the data base is

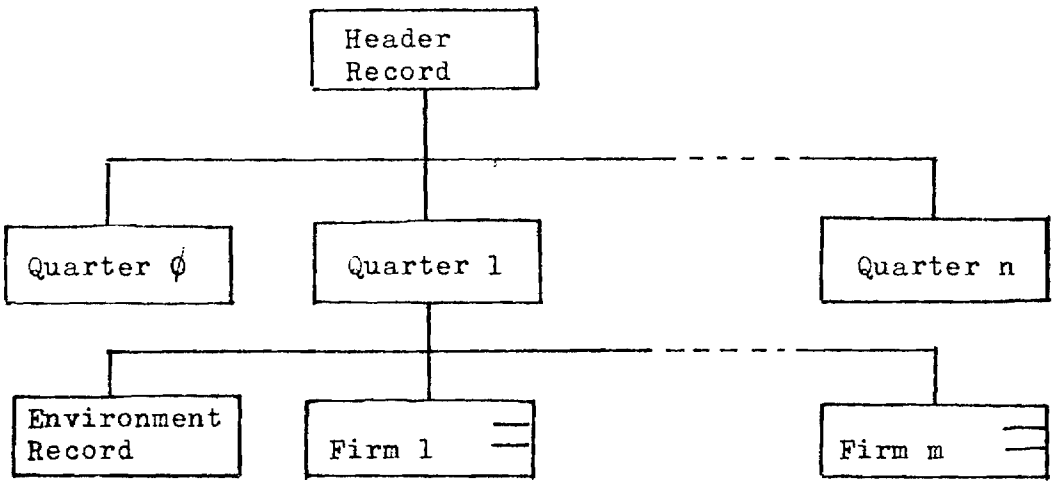


Figure 5.1

Data Base Structure

encrypted, (ICODE), respectively. Level 1 nodes correspond to quarters and physically consist of one 13 value record containing quarterly environmental data. Level 2 nodes correspond to firms and physically consist of three records for each firm: administrative and financial data (37 values) marketing data (33 values) and production data (57 values).

For maximum portability, the data base has been physically organized into a sequential file with the structure shown in Figure 5.1. There will always be one occurrence of the header record, ICQ occurrences of the quarterly environmental records, and, for each quarter, NF occurrences of administrative records, marketing records, and production records. Thus, there will be 1 + ICQ * NF * 3 records in the data base.

Response time and memory requirements can be reduced by using direct access. Unfortunately the syntax for such systems varies among compliers. However, it is relatively easy to convert SLIM to the use of direct access files (this can be a good problem a assignment in data base classes).

5.3 Data Access Mechanism

The procedure for accessing values in the data base is flowcharted in Appendix B. This procedure consists of the following steps:

1. Subroutine LEXAN determines the firm (IFIRM), and quarter

(IQTR) for the requested data item. Also, the first index (INDX) and second index (JNDX) are set for multi-valued data items (defined below).
2. GET is called. If data for the requested quarter is not in main memory, LOAD is called to input the correct quarter.
3. An index (INDEX) into array DB is computed to select the requested data value and the value is stored in both VAL and the next sequential location of TSTORE (starting at 82).

Multi-valued data items are those which may be subscripted with P, A, M, or S (in addition to Q and F). These items will have more than one value per firm, per quarter. Price is an example because it can be subscripted by product and area so each firm will have four prices per quarter.

Index limits for multi-valued data items are given in columns 8 and 9 of the data dictionary. Column 10 indicates the type of the first subscript. The following codes are used:

3 - area only
4 - product, area
5 - material only
6 - stage, area

The second subscript type, if it exists, is always area.

5.4 Structure of the SLIM Program

The general steps in query processing are:

1. Solicit and verify user ID, load data dictionary, initialize variables.
2. Solicit query, perform lexical and syntactic analysis putting data in array TSTORE, codes and pointers to TSTORE in array ISYNTX. Functions are also evaluated at this time.
3. Call appropriate routine to execute command (END, COMPUTE, LIST, PRINT, SUBSET, OPTION, etc).
4. Return to step 2.

As implied in step 2, any data (constants, data item values) are retrieved during the lexical analysis and put into an array

called TSTORE. Specific locations in TSTORE are reserved for each type of value (temporary variable, constant, data base item, function). These reserved locations are given in Table 5.2. For example, the current value of temporary variable A will always be stored in TSTORE (1). The first data base value retrieved will always be in TSTORE (82).

Figure 5.2

Contenta of Array TSTORE

Data Type	TSTORE Position
Temporary Variable (A-W)	1 - 23
Temporary Arrays(X, Y, Z)	24 - 71
Constants	72 - 81
Data Base Item	82 - 209
Function Value	210 - 219
For expansion	220 - 239

The array ISYNTX will contain (1) codes for characters encountered in parsing the query and (2) pointers into TSTORE. ISYNTX (1) always contains the command (key-word) type. Parsing codes are given in Table 5.3.

Figure 5.3

Syntax Parsing Codes

SLIM Symbol	Code in ISYNTX
**	-1
*	-2
/	-3
+	-4
-	-5
(	-6
)	-7
=	-8
,	-9
;	-10
space	-11

•	-12
<	-13
>	-14

Figure 5.4

Command Codes

Command Type	Code in ISYNTX
END	0
COMPUTE	1
LIST	2
PRINT	3
SUBSET	4
OPTION	5
LP; (not implemented)	6
TERMINAL	7
BATCH	8
REWIND	9
REMARK	10
WHEN	11

Appendix A

Subroutine MIS

```
      SUBROUTINE MIS                                                    MIS    1
C  SUBROUTINE TO PERMIT EML TO UPDATE/CREATE THE DATA BASE FOR USE      MIS    2
C  WITH THE -- SLIM -- DATA QUERY PROGRAM.                              MIS    3
C                                                                       MIS    4
      DIMENSION TEMP(90)                                                MIS    5
      DIMENSION A(8,2,2),AD(8,2,2),ADEP(8),ADMC(8),AGENT(8,2),AP(8)     MIS    6
      DIMENSION AQUL(8,2,2),AR(8),BANK(8,2),BANKB(8),BILL(8),BNDS(8)    MIS    7
      DIMENSION BOND(8),BOOK(8),CDOL(8),CPER(8),CPER2(8),CS(8),CY(8,2,2)MIS    8
      DIMENSION DH(8),DLTP(8,2,2),DLTP2(8,2,2),DV(8),ESUM(8),EV(8,2,2)  MIS    9
      DIMENSION EXDOL(8),FACT(8),ICST(8),OAGT(8,2),OL(8),OLB(8)         MIS   10
      DIMENSION PARAM(120),PCT(8,2,2),PQQC(8),PR(8,2,2),PT(8,2,2)       MIS   11
      DIMENSION PV(8,2,2),PV2(8,2,2),PX(8,2,2),R(8),RATE(8),RAW(8,2,2)  MIS   12
      DIMENSION QC(8),RINV(8,2,2),RMP(8,2,2),RSUM(8),SCALE(2,2),SHORT(8)MIS   13
      DIMENSION SK(8),SL(8,2,2),SLL(8,2,2),SS(8),SSUM(8),SSX(8)         MIS   14
      DIMENSION STDC(8,2,2),SMD(8),TIC(8),TRAIN(8,2),TTL(10),XS(8)      MIS   15
      DIMENSION XTC(8,2),ZDS(8),ZRM(2),SSS(8,2),ASUM(8),RD(16)          MIS   16
      DIMENSION TRANS(8,2,2),PE(8,2,2),CGS(8),DEP(8),EQ(8),PEX(8)       MIS   17
      DIMENSION VAR(8),BIN(8),COM(8),EARN(8),ECG(8),FCC(8),PXX(8)       MIS   18
      DIMENSION ARN(8),REV(8),SPR(8),SSR(8),TX(8),PCST(8,2,2)           MIS   19
      DIMENSION PDF(8,2,2),ENG(8),CRF(8),SMDC(8),PDIN(8)                MIS   20
      DIMENSION PTL(2,2),AQL(2,2),AGT(8),TM(2,2),ATL(2,2),BACKO(8,2,2)  MIS   21
      DIMENSION FNZ(2,2) ,AVCMP(8,2),SINV(8),DSUM(8)                    MIS   22
      DIMENSION KSP(8,2),PQC(8),KSTI(8),KSR(8),ZRMC(2),ZRMM(2)          MIS   23
      DIMENSION WTEPS(8),TAXB(8,4),SR(8)                                MIS   24
C     ALL BLANK COMMON                                                  MIS   25
      COMMON A,AD,ADEP,ADMC,AGENT,AP,AQUL,AR,ASUM,BANK,BANKB,BILL,BLRT  MIS   26
      COMMON BNDS,BOND,BOOK,CDOL,CPER,CPER2,CS,CY,DH,DISC,DLTP,DLTP2,DV MIS   27
      COMMON ENDEX,ESUM,EV,EXDOL,FACT,FCST,ICST,K2,K3,KNT,NFRMS,NQTR,NYRMIS   28
      COMMON OAGT,OL,OLB,PARAM,PCT,PQQC,PR,PT,PV,PV2,PX,R,RATE,RAW,QC   MIS   29
      COMMON RINV,RMP,RNN,RSUM,SCALE,SHORT,SK,SL,SLL,SNDX,SS,SSUM,SSX   MIS   30
      COMMON STDC,SMD,TIC,TRAIN,TTL,X,XRM,XS,XTC,Y,Z,ZDS,ZRM,SSS        MIS   31
      COMMON TRANS,PE,CGS,DEP,EQ,PEX,VAR,BIN,COM,EARN,ECG,FCC,PXX,ARN   MIS   32
      COMMON REV,SPR,SSR,TX,PCST,AQL,ATL,PTL,TM,AGT,BACKO,PDF,ENG,CRF   MIS   33
      COMMON SMDC,PDIN,LIN,LO1,LO2,FNZ,AVCMP,SINV,DSUM,RD               MIS   34
      COMMON KSP,PQC,KSTI,KSR,ZRMC,ZRMM,TAXB,ENDX2,SR,PRR               MIS   35
      COMMON WTEPS,NCPYI,NCPYC                                          MIS   36
C     USE THIS COMMON WHEN THE MISGAME IS USED.                         MIS   37
      COMMON LI4,LO3                                                    MIS   38
      EQUIVALENCE (DISC,FLMIS)                                          MIS   39
C                                                                       MIS   40
C      E N V I R O N M E N T                                            MIS   41
```

```
C                                                                       MIS   42
      I=0                                                               MIS   43
      IF(FLMIS.NE.0.)GOTO200                                            MIS   44
      READ(LI4)I,INF                                                    MIS   45
      IF(INF.EQ.NFRMS)GOTO200                                           MIS   46
      WRITE(LO1,2000)                                                   MIS   47
 2000 FORMAT(22H0**********************/22H NO. OF FIRMS CONFLICT)      MIS   48
      STOP                                                              MIS   49
  200 I=I+1                                                             MIS   50
C                                                                       MIS   51
C     FILE FORMAT:                                                      MIS   52
C     RECORD 1: I=NO. PERIODS IN D.B.; NFRMS= NO. FIRMS IN D.B. FILE    MIS   53
C     ENVIRONMENTAL RECORD 'A' FOR START-UP PERIOD                      MIS   54
C     3 RECORDS, 'B', 'C', 'D', FOR EACH FIRM IN START-UP               MIS   55
C     THE RECORDS A,B,C,D ARE REPEATED AS NEEDED FOR THE                MIS   56
C     NUMBER OF PERIODS AND FIRMS IN THIS D.B.                          MIS   57
C                                                                       MIS   58
      WRITE(LO3)I,NFRMS                                                 MIS   59
      IF(I.EQ.1) GO TO 13                                               MIS   60
      NQOLD=I-1                                                         MIS   61
      IF(I.LT.17) GO TO 11                                              MIS   62
C                                                                       MIS   63
C     DATA BASE HOLDS 16 QTRS MAX, DROP OLDEST IF NEEDED.               MIS   64
C     IF DATA BASE STORAGE IS NO PROBLEM, THIS SECTION MAY              MIS   65
C     BE MODIFIED TO STORE ALL PERIODS.  THIS CODE SKIPS N              MIS   66
C     N PERIODS TO LIMIT TOTAL TO 16.                                   MIS   67
C                                                                       MIS   68
      READ(LI4)                                                         MIS   69
      DO 10 J=1,NFRMS                                                   MIS   70
      READ(LI4)                                                         MIS   71
      READ(LI4)                                                         MIS   72
      READ(LI4)                                                         MIS   73
   10 CONTINUE                                                          MIS   74
      NQOLD=15                                                          MIS   75
C                                                                       MIS   76
C     COPY ALL OLD PERIODS TO NEW FILE TO UPDATE D.B.                   MIS   77
C     RECORD 'A' IN QTR TO BE COPIED                                    MIS   78
C                                                                       MIS   79
   11 READ(LI4)(TEMP(N),N=1,13)                                         MIS   80
      WRITE(LO3)(TEMP(N),N=1,13)                                        MIS   81
      DO 500 J=1,NFRMS                                                  MIS   82
```

```
C                                                                       MIS   83
C     RECORD 'B' IN QTR TO BE COPIED                                    MIS   84
      READ(LI4)(TEMP(N),N=1,37)                                         MIS   85
      WRITE(LO3)(TEMP(N),N=1,37)                                        MIS   86
C                                                                       MIS   87
C     RECORD 'C' IN QTR TO BE COPIED                                    MIS   88
      READ(LI4)(TEMP(N),N=1,33)                                         MIS   89
      WRITE(LO3)(TEMP(N),N=1,33)                                        MIS   90
C                                                                       MIS   91
C     RECORD 'D' IN QTR TO BE COPIED                                    MIS   92
      READ(LI4)(TEMP(N),N=34,90)                                        MIS   93
      WRITE(LO3)(TEMP(N),N=34,90)                                       MIS   94
  500 CONTINUE                                                          MIS   95
      NQOLD=NQOLD-1                                                     MIS   96
      IF(NQOLD.GT.0) GO TO 11                                           MIS   97
C                                                                       MIS   98
C     ALL INTEGERS NEED TO BE FLOATED                                   MIS   99
   13 RNQTR=FLOAT(NQTR)                                                 MIS  100
      RNYR=FLOAT(NYR)                                                   MIS  101
C                                                                       MIS  102
C     STORE BILL RATE AS ANNUAL RATE                                    MIS  103
      Y=BLRT*4.                                                         MIS  104
C                                                                       MIS  105
C     NEW RECORD 'A':   13 ITEMS   ************************************MIS  106
      WRITE(LO3) RNQTR,RNYR,ENDEX,ENDX2,FCST,SNDX,(ZRMM(I),I=1,2),      MIS  107
     1(ZRMC(J),J=1,2),(ZRM(I),I=1,2),Y                                  MIS  108
C                                                                       MIS  109
C    F I R M   D A T A                                                  MIS  110
C                                                                       MIS  111
      DO 100 I=1,NFRMS                                                  MIS  112
C                                                                       MIS  113
C        A D MI N I S T R A T I V E   D A T A                           MIS  114
C                                                                       MIS  115
C     ***************************************************************   MIS  116
C     *   NOTE:  SOME SYSTEMS MAY NOT ACCEPT THE COMMENTS IN      *     MIS  117
C     *   THE LOCATIONS BELOW (BETWEEN CONTINUATIONS OF THE       *     MIS  118
C     *   WRITE STATEMENTS.)   IF THIS IS A PROBLEM, SIMPLY       *     MIS  119
C     *   MOVE THE COMMENTS TO BE AHEAD OF THE WRITE.             *     MIS  120
C     ***************************************************************   MIS  121
C                                                                       MIS  122
                                                                        MIS  123
```

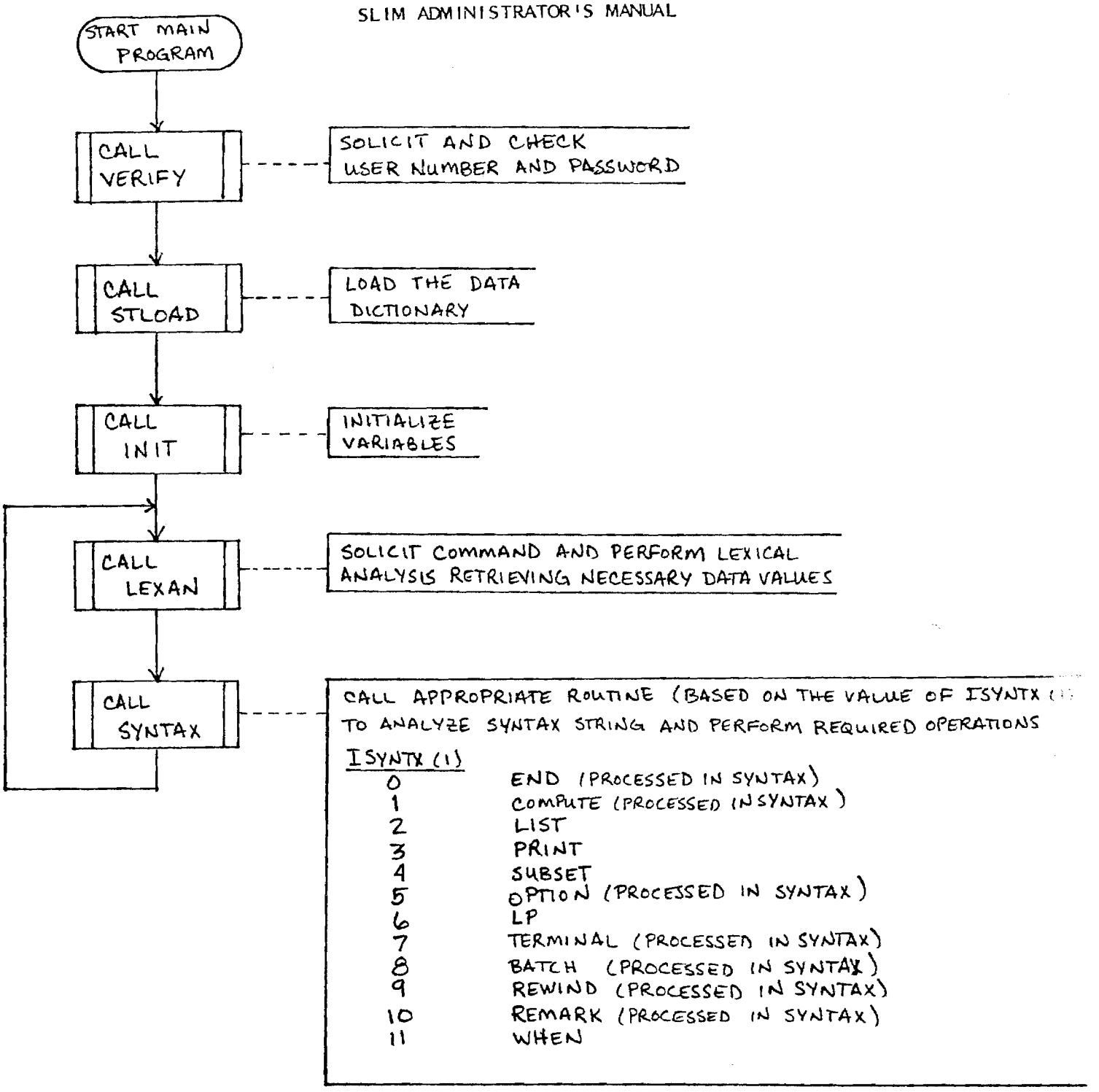
```
C                    INCOME STATEMENT RELATED                                MIS  124
C     NEW RECORD 'B':   37 ITEMS   ***************************************MIS  125
C     KSTI OFFSET IN BML MUST BE REVERSED FOR REPORTING                      MIS  126
C                                                                            MIS  127
      Y=KSTI(I)+1                                                            MIS  128
      WRITE(LO3) REV(I),CGS(I),EXDOL(I),XS(I),TX(I),DEP(I),                  MIS  129
     1EARN(I),ADMC(I),PEX(I),                                                MIS  130
C                    BALANCE SHEET DATA                                      MIS  131
     1CS(I),AR(I),BOOK(I),EQ(I),AP(I),                                       MIS  132
C                    OWNER'S EQUITY                                          MIS  133
     1PDIN(I),SSX(I),SS(I),SSR(I),SPR(I),SK(I),DV(I),                        MIS  134
C                    BONDS                                                   MIS  135
     1BNDS(I),BOND(I),ZDS(I),                                                MIS  136
C                    SHORT TERM INVESTMENTS                                  MIS  137
     1SINV(I),BILL(I),Y,FCC(I),FACT(I),                                      MIS  138
C                    CREDIT RATING AND LOANS                                 MIS  139
     1RATE(I),BIN(I),(BANK(I,J),J=1,2),OLB(I),SHORT(I),SR(I),OL(I)           MIS  140
                                                                             MIS  141
C                                                                            MIS  142
C      M A R K E T I N G   D A T A                                           MIS  143
C                                                                            MIS  144
C      (ORGANIZATION OF ALL FIRM DATA DEALING WITH PRODUCT                   MIS  145
C      AND AREA IS ORGANIZED (<I POSITION>,AREA) WHERE AREA                  MIS  146
C      VARIES MORE SLOWLY THAN <I POSITION>) <I POSITION> IS PROD,           MIS  147
C      MATERIAL, OR STAGE                                                    MIS  148
C                                                                            MIS  149
C     KSP ARRAY NEEDS TO BE FLOATED FOR DATA BASE                            MIS  150
      W=KSP(I,1)                                                             MIS  151
      X=KSP(I,2)                                                             MIS  152
C     NEW RECORD 'C':   33 ITEMS   ***************************************MIS  153
      WRITE(LO3) ((SL(I,K,J),K=1,2),J=1,2),((PR(I,K,J),K=1,2),J=1,2),        MIS  154
     1((PCT(I,K,J),K=1,2),J=1,2),                                            MIS  155
C                    ADVERTISING                                             MIS  156
     1((AD(I,K,J),K=1,2),J=1,2),                                             MIS  157
C                    SALES FORCE                                             MIS  158
     1(OAGT(I,J),J=1,2),TRAIN(I,1),COM(I),CDOL(I),                           MIS  159
     1CPER(I),CPER2(I),                                                      MIS  160
C                    SALES LOST & SHIPPING POLICY                            MIS  161
     1((SLL(I,K,J),K=1,2),J=1,2),W,X,                                        MIS  162
C                     QUALITY CONTROL AND R&D                                MIS  163
     1QC(I),PQC(I),RD(I),RD(I+8)                                             MIS  164
```

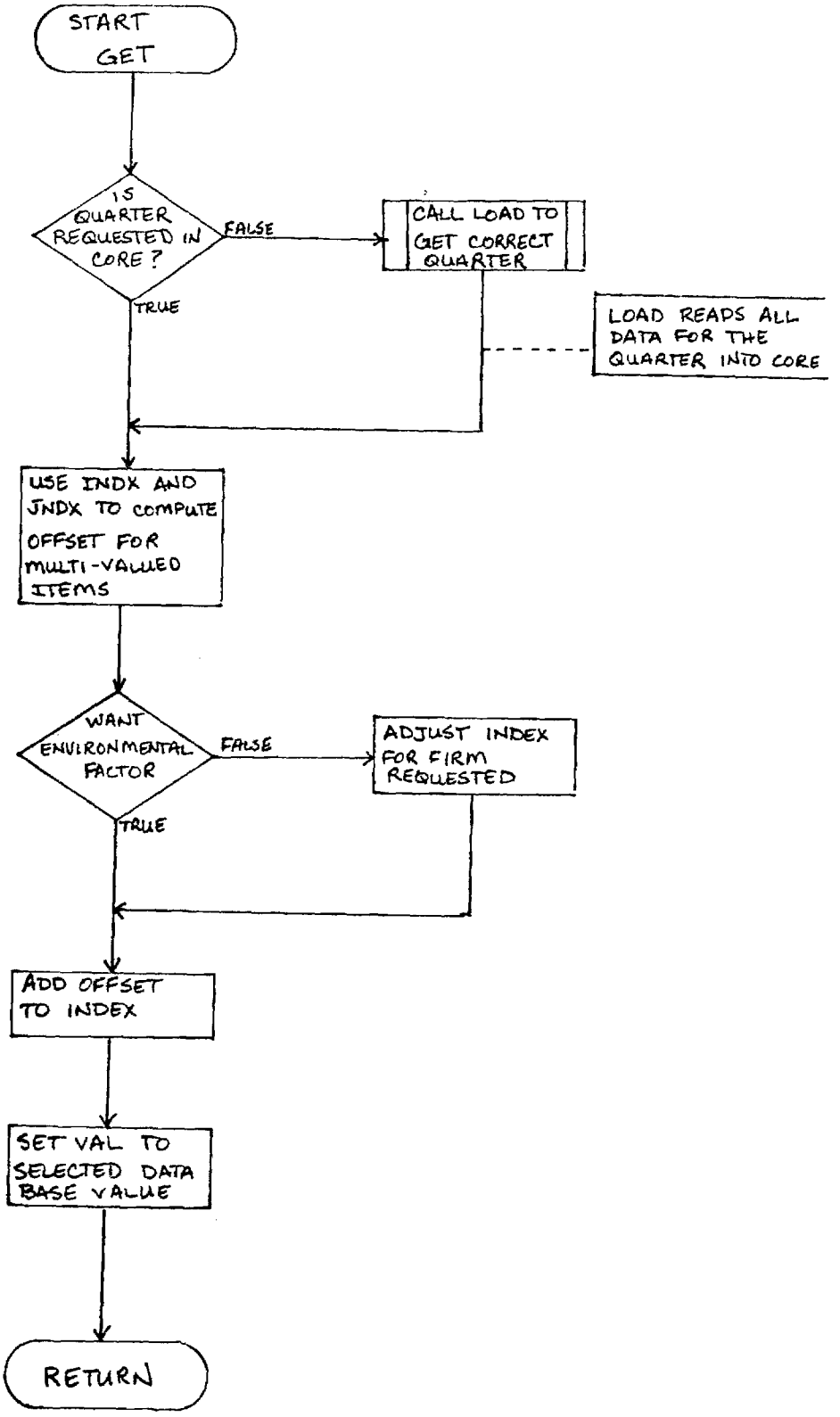
```
C                                                                       MIS  165
C       P R O D U C T I O N                                             MIS  166
C                                                                       MIS  167
C     NEW RECORD 'D':   57 ITEMS   ***************************************MIS  168
C                     CAPACITY                                          MIS  169
      WRITE(LO3)                                                        MIS  170
     1((PV(I,K,J),K=1,2),J=1,2),((CY(I,K,J),K=1,2),J=1,2),              MIS  171
     1((DLTP(I,K,J),K=1,2),J=1,2),((PX(I,K,J),K=1,2),J=1,2),            MIS  172
C                     INVENTORY                                         MIS  173
     1((EV(I,K,J),K=1,2),J=1,2),((BACKO(I,K,J),K=1,2),J=1,2),           MIS  174
     1((RINV(I,K,J),K=1,2),J=1,2),((RAW(I,K,J),K=1,2),J=1,2),           MIS  175
     1((TRANS(I,K,J),K=1,2),J=1,2),ECG(I),                              MIS  176
     1((RMP(I,J,K),J=1,2),K=1,2),((STDC(I,J,K),J=1,2),K=1,2),           MIS  177
C                     PRODUCTION COSTS                                  MIS  178
     1((PCST(I,K,J),K=1,2),J=1,2),ENG(I),CRF(I),                        MIS  179
     1(XTC(I,K),K=1,2),((PV2(I,K,J),K=1,2),J=1,2)                       MIS  180
  100 CONTINUE                                                          MIS  181
      RETURN                                                            MIS  182
      END                                                               MIS  183
```

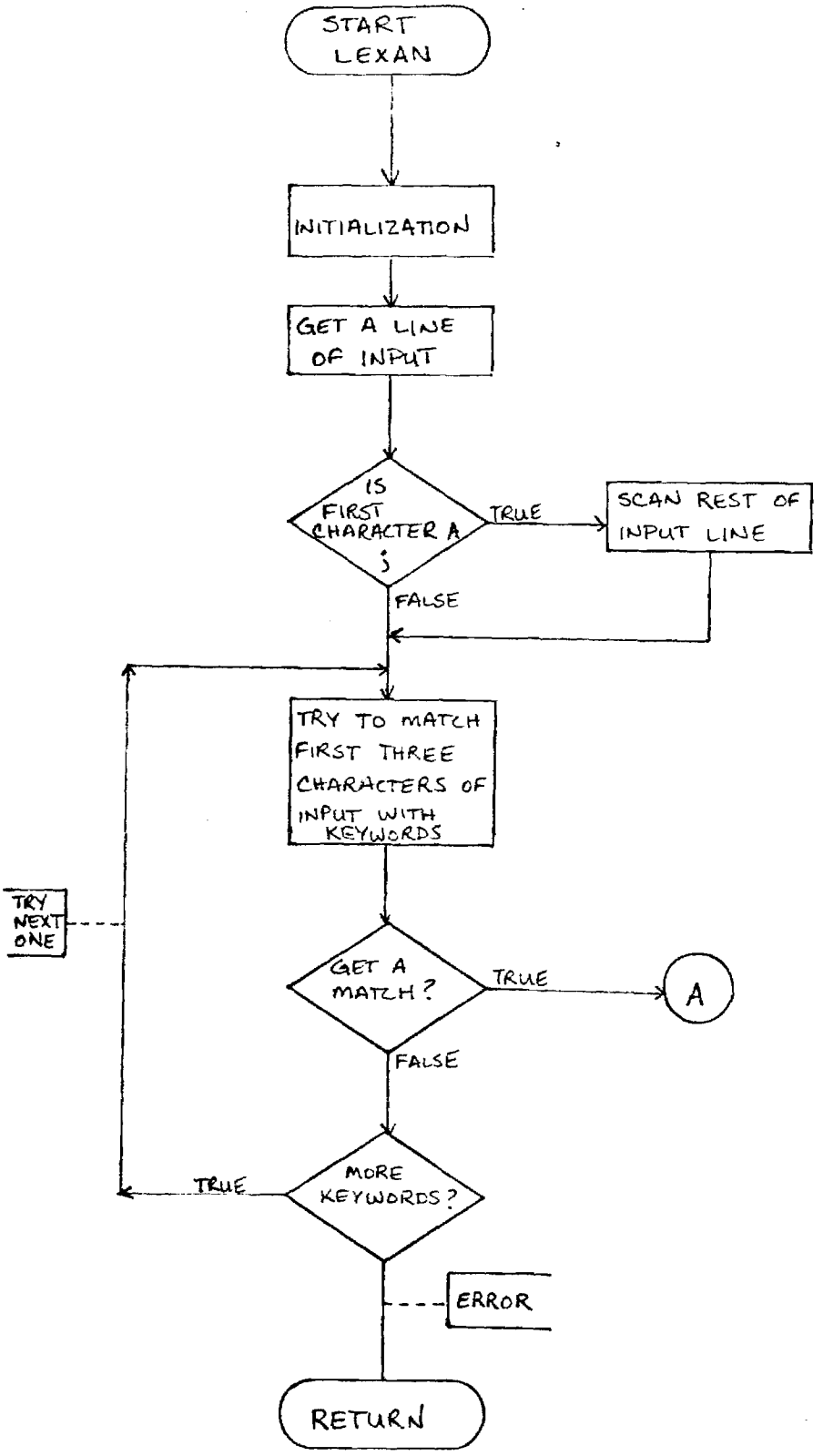
Appendix B

SLIM Flowcharts

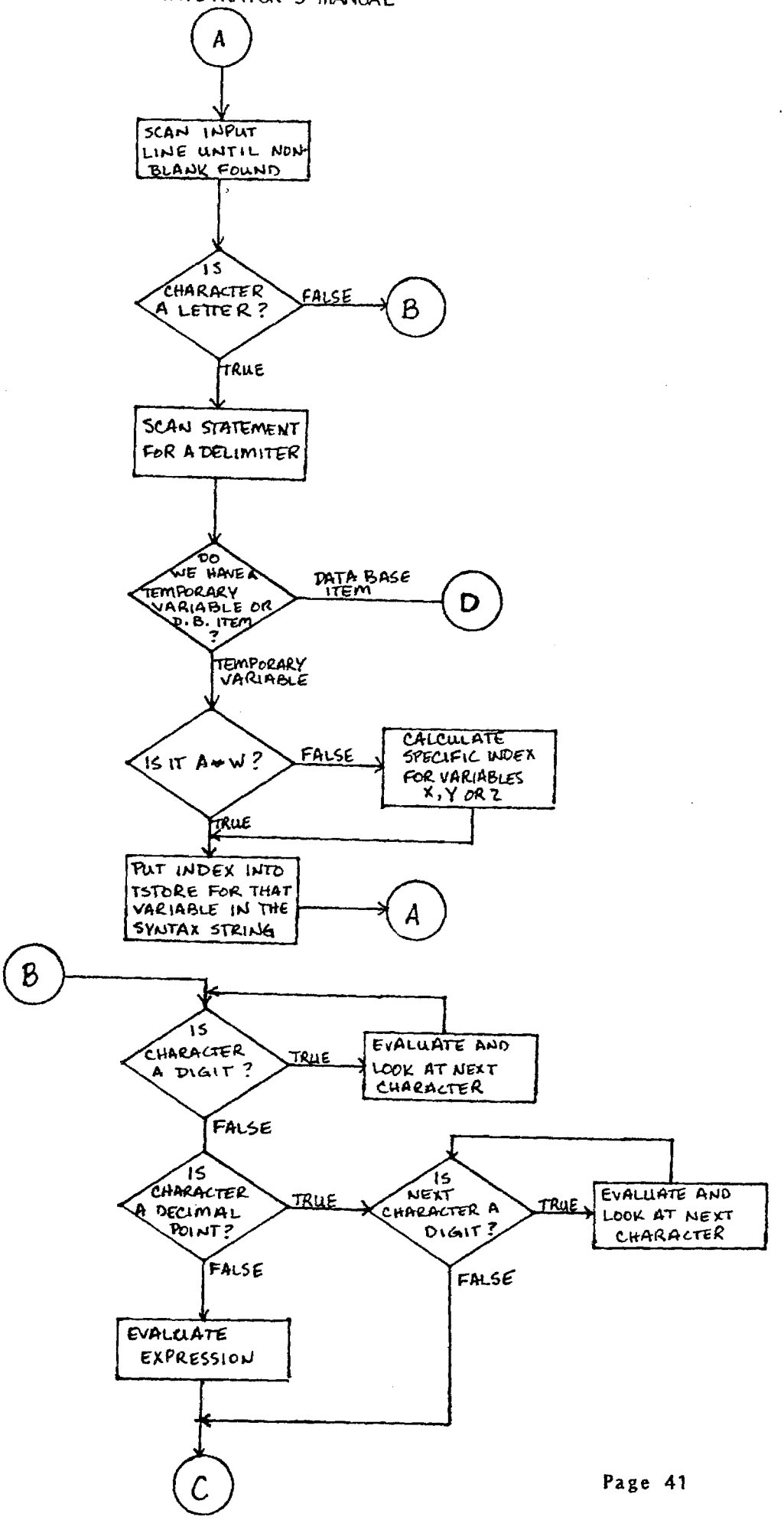
On the following pages certain key portions of SLIM are flowcharted. Other sections which are easily read from the program listing are not included here. The major structure is the LEXAN - SYNTAX loop, where each subroutine calls various service subroutines to perform specific tasks, such as requesting a command from the terminal, computing a sum of several items in TSTORE, etc.


START MAIN PROGRAM
CALL VERIFY
SOLICIT AND CHECK USER NUMBER AND PASSWORD
CALL STLOAD
LOAD THE DATA DICTIONARY
CALL INIT
INITIALIZE VARIABLES
CALL LEXAN
SOLICIT COMMAND AND PERFORM LEXICAL ANALYSIS RETRIEVING NECESSARY DATA VALUES
CALL SYNTAX
CALL APPROPRIATE ROUTINE (BASED ON THE VALUE OF ISYNTX (1)
TO ANALYZE SYNTAX STRING AND PERFORM REQUIRED OPERATIONS
ISYNTX (1)
0 END (PROCESSED IN SYNTAX)
1 COMPUTE (PROCESSED IN SYNTAX)
2 LIST
3 PRINT
4 SUBSET
5 OPTION (PROCESSED IN SYNTAX)
6 LP
7 TERMINAL (PROCESSED IN SYNTAX)
8 BATCH (PROCESSED IN SYNTAX)
9 REWIND (PROCESSED IN SYNTAX)
10 REMARK (PROCESSED IN SYNTAX)
11 WHEN

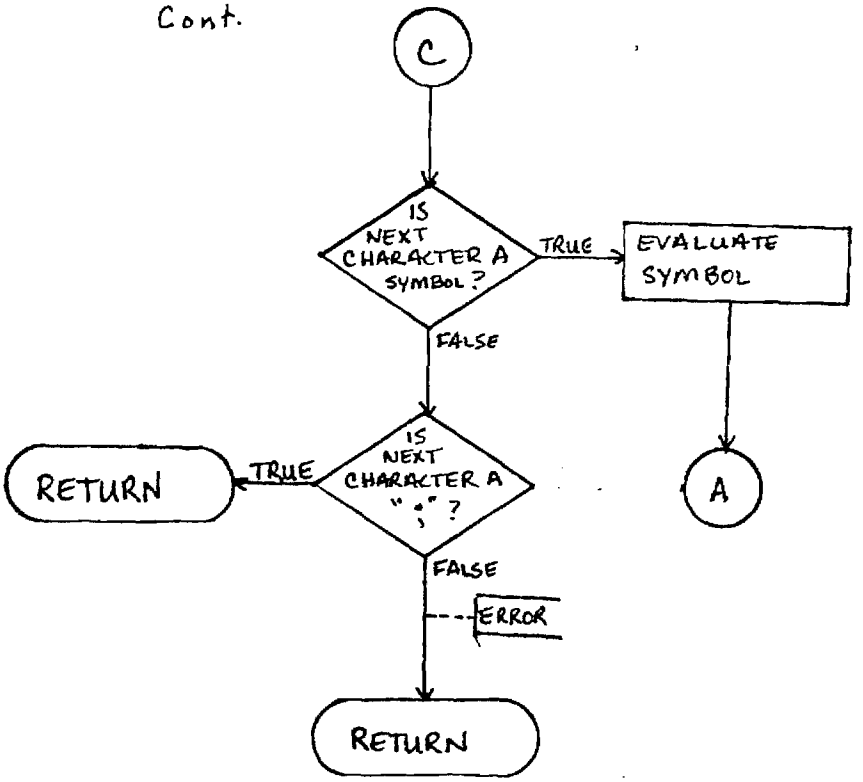

START
GET
IS QUARTER REQUESTED IN CORE ?
FALSE
TRUE
CALL LOAD TO GET CORRECT QUARTER
LOAD READS ALL DATA FOR THE QUARTER INTO CORE
USE INDX AND JNDX TO COMPUTE OFFSET FOR MULTI-VALUED ITEMS
WANT ENVIRONMENTAL FACTOR
FALSE
TRUE
ADJUST INDEX FOR FIRM REQUESTED
ADD OFFSET TO INDEX
SET VAL TO SELECTED DATA BASE VALUE
RETURN

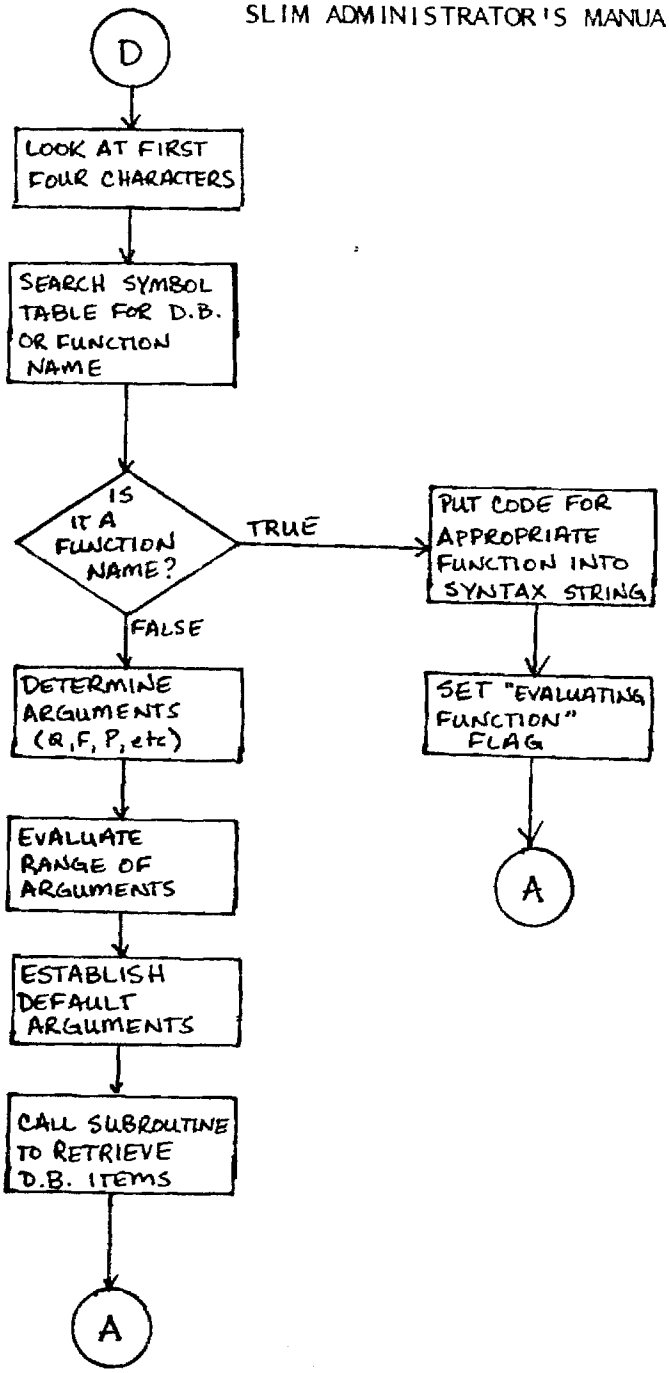

START LEXAN
INITIALIZATION
GET A LINE OF INPUT
IS FIRST CHARACTER A ;
TRUE
SCAN REST OF INPUT LINE
FALSE
TRY TO MATCH FIRST THREE CHARACTERS OF INPUT WITH KEYWORDS
TRY NEXT ONE
GET A MATCH ?
TRUE
A
FALSE
MORE KEYWORDS ?
TRUE
ERROR
RETURN

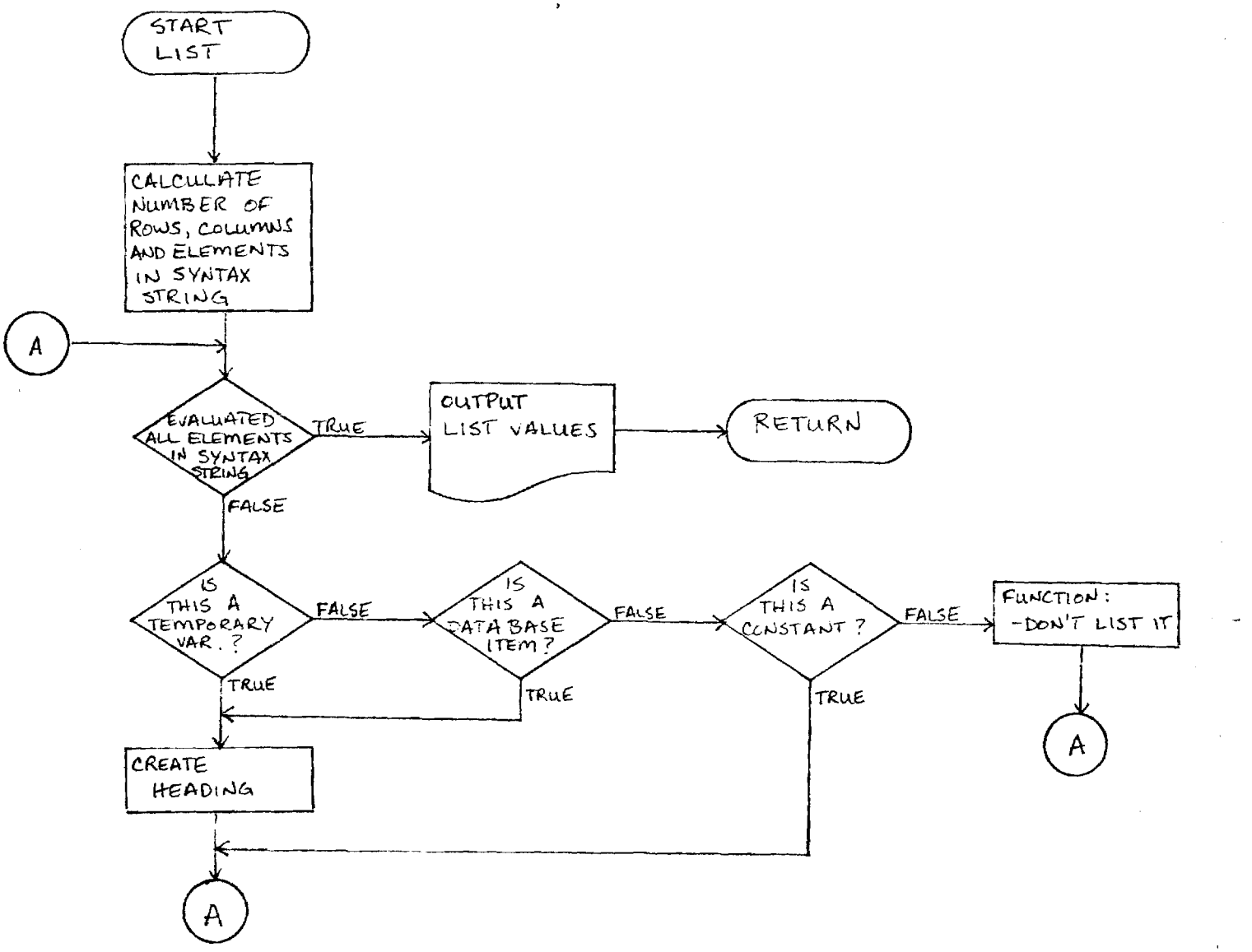
LEXAN
Cont.

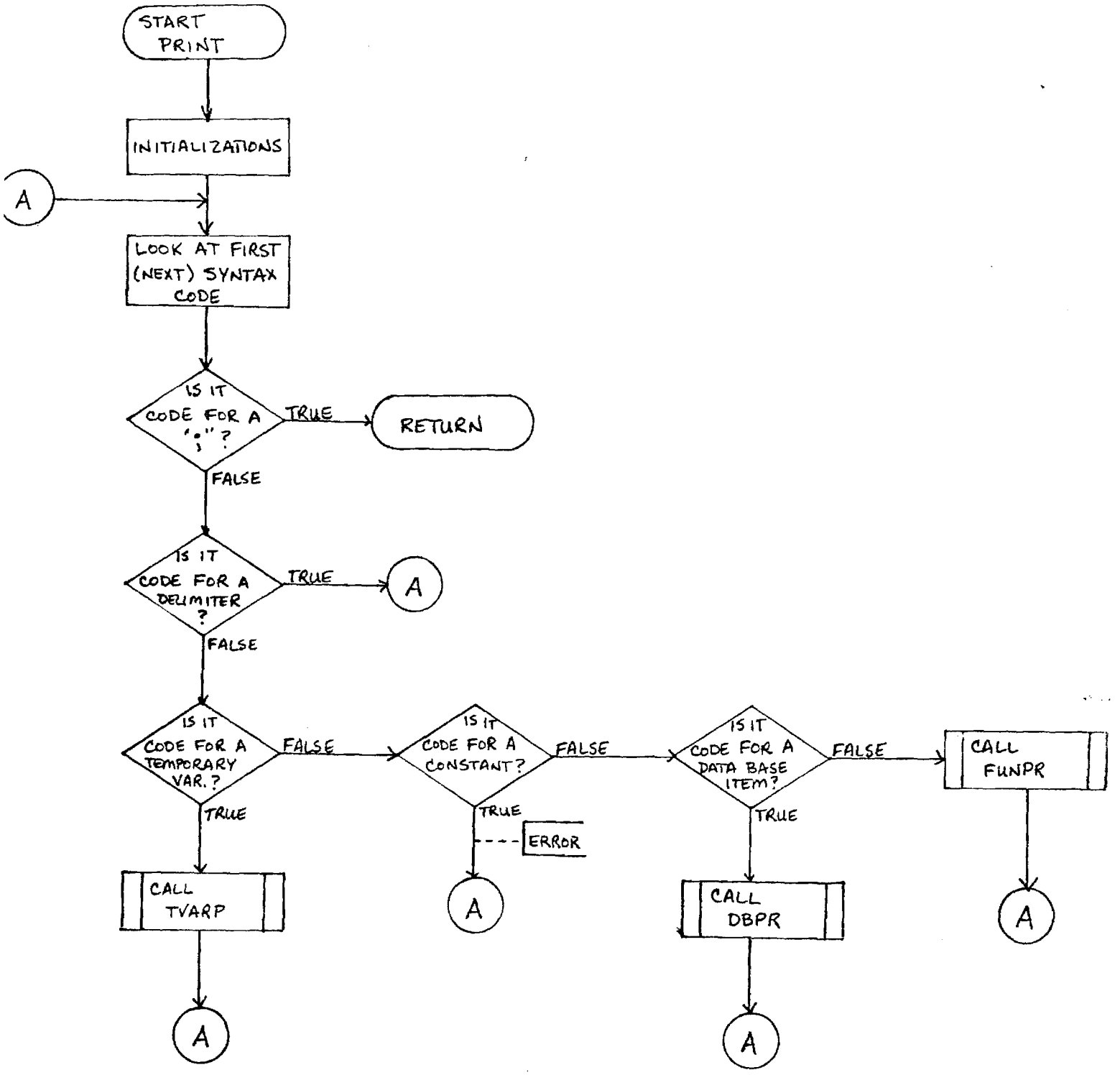

A
SCAN INPUT LINE UNTIL NON-BLANK FOUND
IS CHARACTER A LETTER?
FALSE
B
TRUE
SCAN STATEMENT FOR A DELIMITER
DO WE HAVE A TEMPORARY VARIABLE OR D.B. ITEM ?
DATA BASE ITEM
D
TEMPORARY VARIABLE
IS IT A=W?
FALSE
CALCULATE SPECIFIC INDEX FOR VARIABLES X, Y OR Z
TRUE
PUT INDEX INTO TSTORE FOR THAT VARIABLE IN THE SYNTAX STRING
A
B
IS CHARACTER A DIGIT ?
TRUE
EVALUATE AND LOOK AT NEXT CHARACTER
FALSE
IS CHARACTER A DECIMAL POINT?
TRUE
IS NEXT CHARACTER A DIGIT?
TRUE
EVALUATE AND LOOK AT NEXT CHARACTER
FALSE
FALSE
EVALUATE EXPRESSION
C

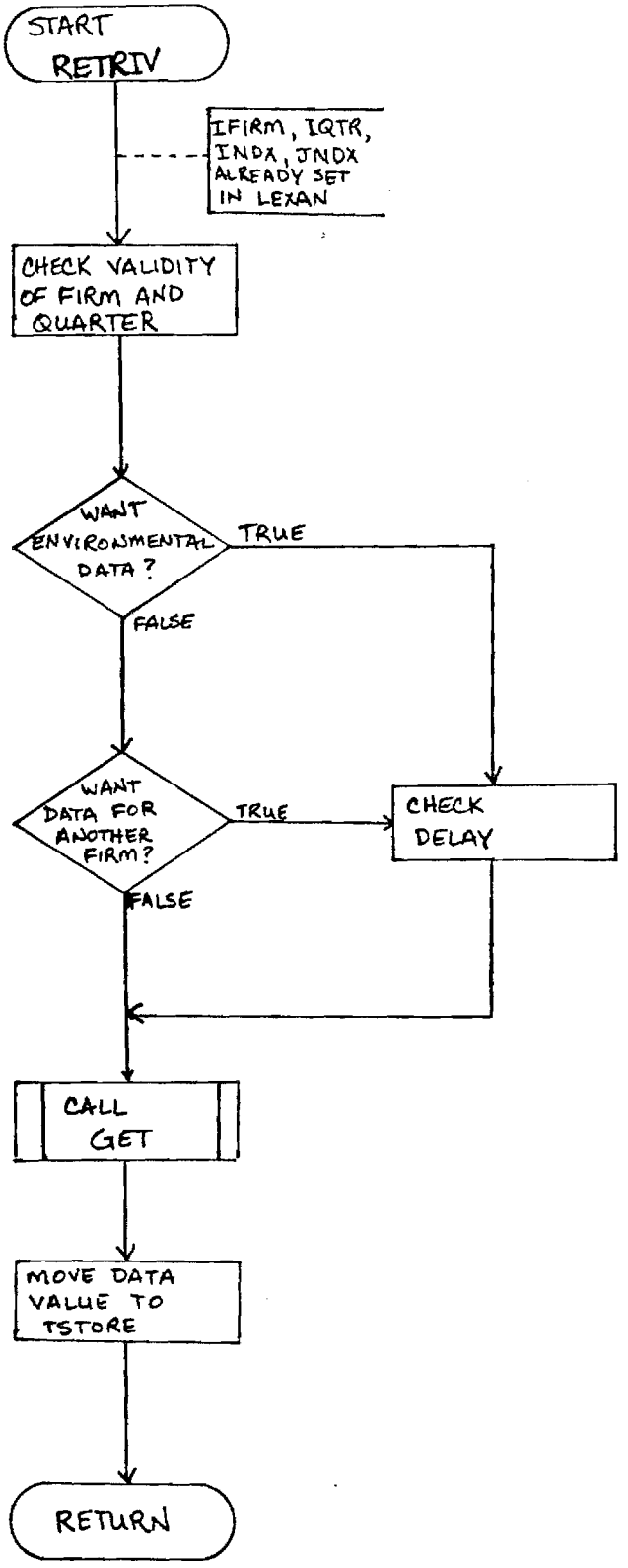
LEXAN
Cont.

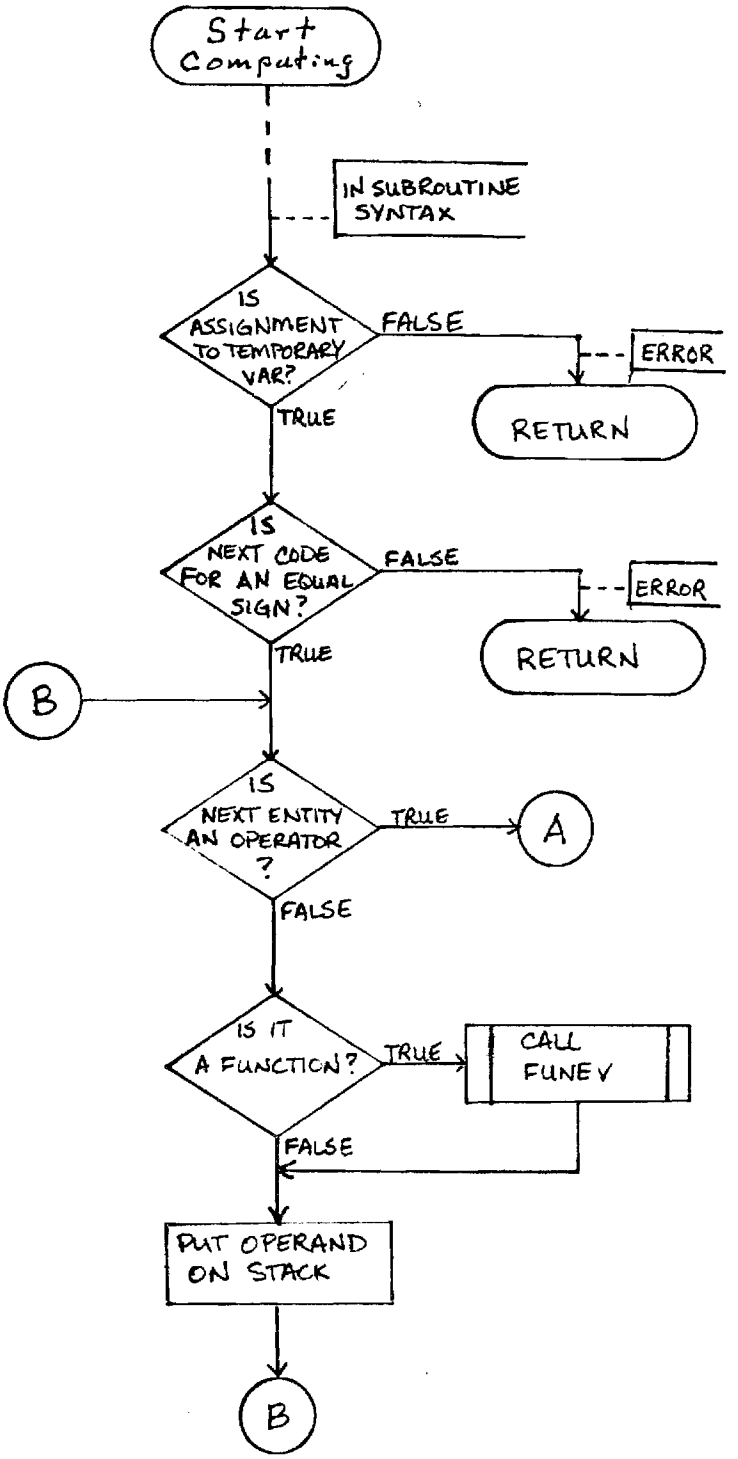

C
IS NEXT CHARACTER A SYMBOL?
TRUE
EVALUATE SYMBOL
FALSE
A
IS NEXT CHARACTER A ";" ?
TRUE
RETURN
FALSE
ERROR
RETURN

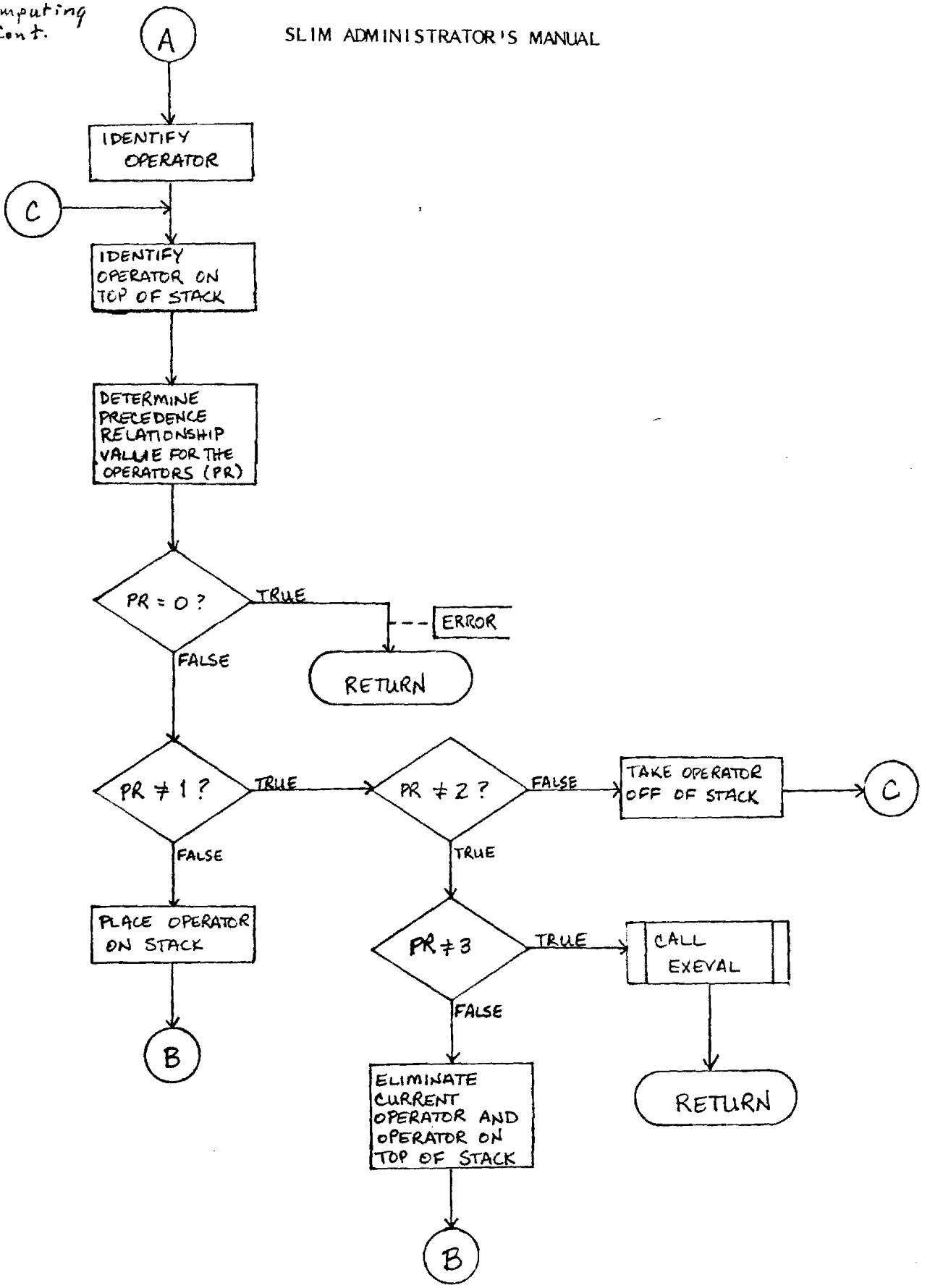

D
LOOK AT FIRST FOUR CHARACTERS
SEARCH SYMBOL TABLE FOR D.B. OR FUNCTION NAME
IS IT A FUNCTION NAME?
TRUE
FALSE
PUT CODE FOR APPROPRIATE FUNCTION INTO SYNTAX STRING
SET "EVALUATING FUNCTION" FLAG
A
DETERMINE ARGUMENTS (R, F, P, etc)
EVALUATE RANGE OF ARGUMENTS
ESTABLISH DEFAULT ARGUMENTS
CALL SUBROUTINE TO RETRIEVE D.B. ITEMS
A

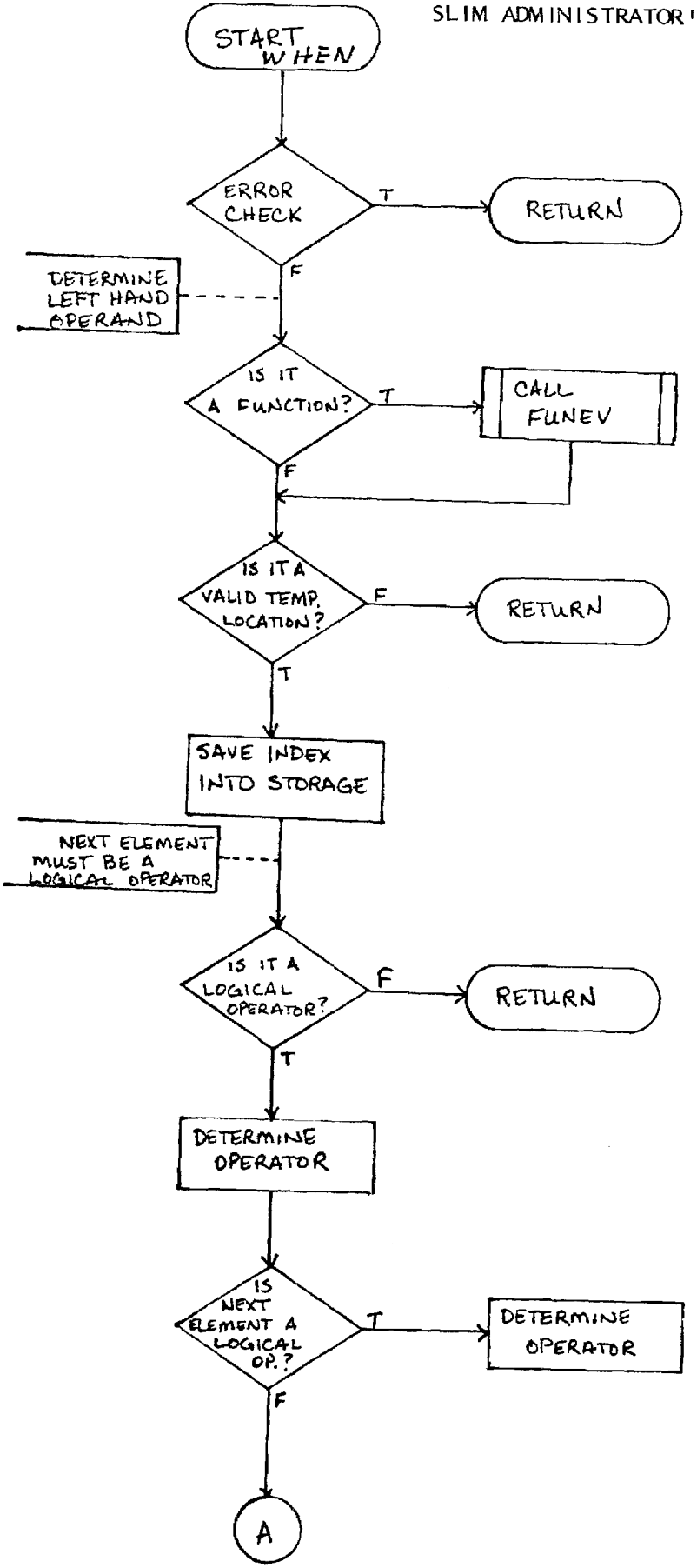

START LIST
CALCULATE NUMBER OF ROWS, COLUMNS AND ELEMENTS IN SYNTAX STRING
A
EVALUATED ALL ELEMENTS IN SYNTAX STRING
TRUE
OUTPUT LIST VALUES
RETURN
FALSE
IS THIS A TEMPORARY VAR.?
FALSE
IS THIS A DATA BASE ITEM?
FALSE
IS THIS A CONSTANT?
FALSE
FUNCTION: -DON'T LIST IT
TRUE
TRUE
TRUE
A
CREATE HEADING
A


START PRINT
INITIALIZATIONS
A
LOOK AT FIRST (NEXT) SYNTAX CODE
IS IT CODE FOR A ';' ?
TRUE
RETURN
FALSE
IS IT CODE FOR A DELIMITER ?
TRUE
A
FALSE
IS IT CODE FOR A TEMPORARY VAR.?
FALSE
IS IT CODE FOR A CONSTANT?
FALSE
IS IT CODE FOR A DATA BASE ITEM?
FALSE
CALL FUNPR
TRUE
CALL TVARP
A
TRUE
ERROR
A
TRUE
CALL DBPR
A
A

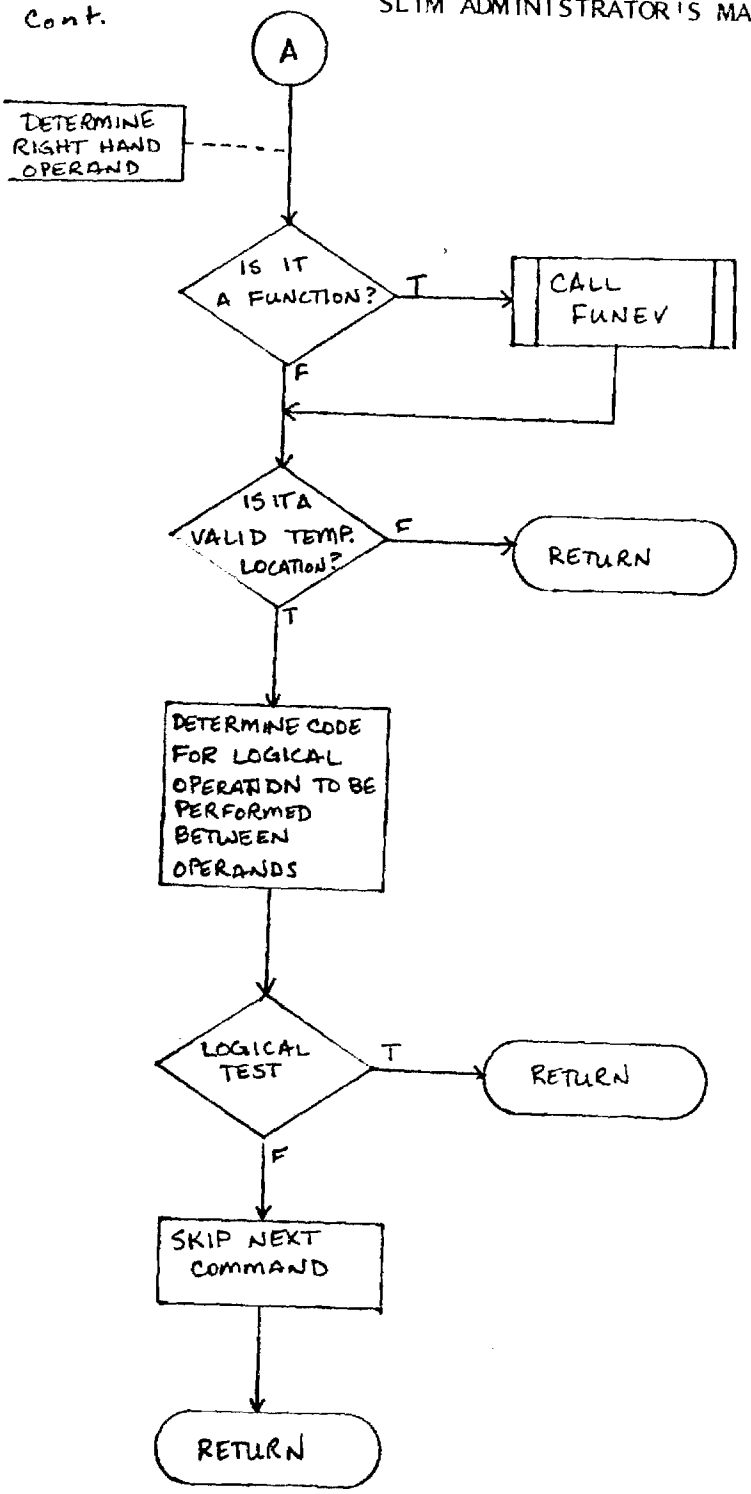

START
RETRIV
IFIRM, IQTR, INDX, JNDX ALREADY SET IN LEXAN
CHECK VALIDITY OF FIRM AND QUARTER
WANT ENVIRONMENTAL DATA ?
TRUE
FALSE
WANT DATA FOR ANOTHER FIRM?
TRUE
CHECK DELAY
FALSE
CALL GET
MOVE DATA VALUE TO TSTORE
RETURN


Start Computing
IN SUBROUTINE SYNTAX
IS ASSIGNMENT TO TEMPORARY VAR?
FALSE
ERROR
RETURN
TRUE
IS NEXT CODE FOR AN EQUAL SIGN?
FALSE
ERROR
RETURN
TRUE
B
IS NEXT ENTITY AN OPERATOR ?
TRUE
A
FALSE
IS IT A FUNCTION?
TRUE
CALL FUNEV
FALSE
PUT OPERAND ON STACK
B


A
IDENTIFY OPERATOR
C
IDENTIFY OPERATOR ON TOP OF STACK
DETERMINE PRECEDENCE RELATIONSHIP VALUE FOR THE OPERATORS (PR)
PR = 0 ?
TRUE
ERROR
RETURN
FALSE
PR ≠ 1 ?
TRUE
PR ≠ 2 ?
FALSE
TAKE OPERATOR OFF OF STACK
C
FALSE
TRUE
PLACE OPERATOR ON STACK
B
PR ≠ 3
TRUE
CALL EXEVAL
FALSE
RETURN
ELIMINATE CURRENT OPERATOR AND OPERATOR ON TOP OF STACK
B


START WHEN
ERROR CHECK
T
RETURN
F
DETERMINE LEFT HAND OPERAND
IS IT A FUNCTION?
T
CALL FUNEV
F
IS IT A VALID TEMP. LOCATION?
F
RETURN
T
SAVE INDEX INTO STORAGE
NEXT ELEMENT MUST BE A LOGICAL OPERATOR
IS IT A LOGICAL OPERATOR?
F
RETURN
T
DETERMINE OPERATOR
IS NEXT ELEMENT A LOGICAL OP.?
T
DETERMINE OPERATOR
F
A

WHEN Cont.


A
DETERMINE RIGHT HAND OPERAND
IS IT A FUNCTION?
T
CALL FUNEV
F
IS IT A VALID TEMP. LOCATION?
F
RETURN
T
DETERMINE CODE FOR LOGICAL OPERATION TO BE PERFORMED BETWEEN OPERANDS
LOGICAL TEST
T
RETURN
F
SKIP NEXT COMMAND
RETURN

Appendix C

Variable Definitions

DB (1030): Data Base-One Quarter's Worth of Data

DIGIT (10): Array of Values 0-9

I: Indexes Line (input line)

IACT (4,10): Pointer to Subscript type for Subscript I, data item J.

IALPHA (26,2): Array of Alphabetic Characters and Corresponding Indexes. Pointers into TSTORE. Col. 1 contains the name in A1 format (A-Z), Col. 2 contains the pointer (A=1, B=2,...,X=24, Y=40, Z=56).

IANDX (6): Used only in BLDSUB as a pointer into IALPHA.
- IANDX(1) = 6 F (Firm)
- IANDX(2) = 17 Q (Quarter)
- IANDX(3) = 1 A (Area)
- IANDX(4) = 16 P (Product)
- IANDX(5) = 13 M (Material)
- IANDX(6) = 19 S (Stage)

IAQ: Quarter which is currently in main memory

ICODE: Flag in data base header record indicating whether data base is encrypted. (0 = not encrypted)

IDABSR: Data base file

IDATBS: Optional debug file, not referenced in current version.

IDBT: Pointer into TSTORE to show where the next data base value should be stored.

ICQ: Current Quarter

IDEF (6): Default values (set in INIT)
- IDEF(1): Default firm, defaults to current
- IDEF(2): Default quarter, defaults to current
- IDEF(3-6): Default area, product, material, stage

IDEL: Delay associated with the current data item

IDIGIT (10): Array of Numeric Characters 0-9

IENT: Counter for number of data items processed in a command (Max of 10)

IFILE: Subset output file

IFIRM: Firm Number for which data is requested for the item being processed. Equivalenced to IRET(1) in LEXAN.

IFIRST: Indicates legal subscript types. Read from colum 10 of the data dictionary. The following codes are used:
3-area only
4-product,area
5-material only
6-stage,area

IFQ: Number of first quarter. In the current version this will always be 1.

IFR(4,10): First value in range for subscript I of data item J encountered in command. LST(I,J) is last value in range.

IFUN: Location in TSTORE at which functions start, currently 210.

IHEAD (6,4): Used to Store Headings for Output ("List" "Compute" Commands)

INDEX: Computed by GET to select the data base value requested from array DB.

INDX: First subscript in offset for multi-valued data items.

IOPT: Code for OPTION command 0=No options; 1=sum columns, 2=average columns

IOUT (70): Line of output for PRINT command

IQTR: Quarter Number for which data is requested in item being processed. Equivalenced to IRET(2) in LEXAN.

IPOL (72): Array used to Build Polish Notation String For "Compute" Command. Also used to store LIST subscipts before output.

IRAN: Percent random error associated with the current data item

ISKP: Number of quarters to skip when reading database file.

ISTL: Index used in searching the data dictionary for a symbol match.

ISTLOC (10): Pointer to the line in the symbol for the ith data base item encountered in the command.

ISYM(13,2): Array of delimiters and corresponding codes. Col. 1 contains the characters in A1 format, read from the data dictionary file. Col. 2 contains numeric codes (-2,-3,...)

ISYMT (1090): The data dictionary (90 lines max)
Columns 1-4: Data item name
Column 5: Delay
Column 6: Random error
Column 7: Relative location in DB array
Column 8: Max I index for subscript
Column 9: Max J index for subscript
Column 10: Subscript type

ISYMTB: Data dictionary file

ISYNTX (100): Syntax String

ITC: Pointer to SLIM subscript type:
1 = Firm
2 = Quarter
3 = Area
4 = Product
5 = Material
6 = Stage

ITS1 (10): Location in TSTORE where the ith data base item begins.

ITS2 (10): Location in TSTORE where the ith data base item ends.

ITY (4,10): Subscript type (see ITC) for argument i of jth data base item in the query.

IUFN: Current user's firm. Set at login time.

IUIN: Current user's industry

IUSNUM: User Number File-Same as data dictionary

JNDX: Second offset in subscript for multi-valued data items.

K: Indexes array IPOL during processing of COMPUTE commands in SYNTAX

KEY (13,3): without LP version. Array of Keywords (commands). Stored one character per column, A1 format. Thus row one has 3 cols. and 3 characters (E N D), row two has 3 (C O M) etc. All commands are interpreted by the first three characters, with row position identifying the command.

LINE (72): Array for one line of input

LST (4,10): Last value in range for subscript I of data item J encountered in command. IFR(I,J) is first value in range.

MAXI: Maximum value for I subscript type

MAXJ: Maximum value for J subscript type

NAME (4): Used to hold name of data base items (for output of "LISTS" etc.)

NDXREL: Points to the relative location in DB of the data item currently being processed. This must be adjusted for the correct firm and multivalued items such as PRICE.

NENVF: Number of environmental factors

NERR: Error number

NF: Number of Firms.

NFF: Number of Firm Factors.

NFF1: Number of Firm Factors Block 1

NFF2: Number of Firm Factors Block 2.

NFF3: Number of Firm Factor Block 3

NSTEL: Number of Symbol Table Elements

TOTAL (10): Column totals for LIST or SUBSET commands

TRMIN: Unit number for input from terminal

TRMOUT: Unit number for output to terminal

TSTORE (219): Array containing values of:
Temporary scalars (1-23)
Temporary Arrays (24-71)
Constants (72-81)
Database Items (82-209)
Functions (210-219)
Expansion space (220-239)

SUBOUT (13): Line of output (data) for SUBSET and LIST

VAL: Value for data base item

Appendix D

Data Dictionary

```
   6 USERS FOR MIS GAME
00,ADMIN
11,FIRM ONE
12,FIRM TWO
13,FIRM THREE
14,FIRM FOUR
15,FIRM FIVE
16,FIRM SIX
17,FIRM SEVEN
18,FIRM EIGHT
  83 SYMBOL TABLE ELEMENTS
0013 ENVIRONMENTAL FACTORS
0127 FIRM FACTORS
0037 FIRM FACTORS IN BLOCK1
0033 FIRM FACTORS IN BLOCK2
0057 FIRM FACTORS IN BLOCK3
0005 FUNCTION NAMES
```

3NAME	BML	DELAY	%RANDOM	INDEX	ARRAY		IPOS	DESCRIPTION
3	NAME		ERROR	POS.	LIMITS		TYPE	
QTR	NQTR	0	0	1	0	0	0	QUARTER WITHIN YEAR (1, 2, 3, OR 4)
YEAR	NYR	0	0	2	0	0	0	YEAR (77, 78, ETC.)
EI	ENDEX	0	0	3	0	0	0	ECONOMIC INDICATOR
EINQ	ENDX2	0	0	4	0	0	0	ECONOMIC INDICATOR,NEXT QTR
EINY	FCST	0	0	5	0	0	0	ECONOMIC INDICATOR, NEXT YEAR
BBI	SNDX	0	0	6	0	0	0	BULL-BEAR INDEX
PRAW	ZRMM	0	0	7	2	0	5	PRICE OF RAW MATERIALS(MATERIAL)
FUTLQ	ZRMC	0	0	9	2	0	5	LAST QUARTER FUTURES PRICE(MATERIAL)
FUT	ZRM	0	0	11	2	0	5	CURRENT QUARTER FUTURES(MATERIAL)
STBR	BLRT	0	0	13	0	0	0	SHORT TERM BILL RATE
SALES	REV	4	0	14	0	0	0	SALES REVENUE
COGS	CGS	4	10	15	0	0	0	COST OF GOODS SOLD
FC	EXDOL	99	0	16	0	0	0	FIXED COSTS
ME	XS	4	10	17	0	0	0	MISCELLANEOUS EXPENSES
TAX	TX	4	0	18	0	0	0	TAX LIABILITY
DEPR	DEP	4	0	19	0	0	0	DEPRECIATION CHARGE
BL	EARN	0	0	20	0	0	0	NET EARNINGS (BOTTOM LINE) AFTER TAXES
AC	ADMC	4	10	21	0	0	0	ADMINISTRATIVE COSTS
CE	PEX	99	0	22	0	0	0	CAPITAL EXPENDITURES
CASH	CS	4	10	23	0	0	0	CASH ASSETS
AR	AR	4	10	24	0	0	0	ACCOUNTS RECEIVABLE
PLANT	BOOK	4	20	25	0	0	0	BOOK VALUE OF PLANT

OE	EQ	4	0	26	0	0	0	OWNERS' EQUITY
AP	AP	4	10	27	0	0	0	ACCOUNTS PAYABLE
PIC	PDIN	4	0	28	0	0	0	PAID IN CAPITAL
SOUT	SSX	0	0	29	0	0	0	NUMBER OF SHARES OUTSTANDING
SOFF	SS	0	0	30	0	0	0	NUMBER OF SHARES OFFERED FOR SALE
SSR	SSR	0	0	31	0	0	0	STOCK SALE RECEIPTS
SSP	SPR	0	0	32	0	0	0	STOCK SALES PRICE
SP	SK	0	0	33	0	0	0	STOCK PRICE FROM PREVIOUS QUARTER
DIVDCL	DV	0	0	34	0	0	0	DIVIDENDS DECLARED, CURRENT QUARTER
FVBO	BNDS	0	0	35	0	0	0	CHANGE IN FACE VALUE OF BONDS (SALE OR CALL)
BOUT	BOND	0	0	36	0	0	0	OUTSTANDING BOND ISSUES
UBD	ZDS	0	0	37	0	0	0	UNAMORTIZED BOND DISCOUNT
STII	SINV	0	10	38	0	0	0	SHORT TERM INVESTMENT INCOME
STI	BILL	99	0	39	0	0	0	SHORT TERM(90 DAY) INVSTMENT
IT	KSTI	99	0	40	0	0	0	TYPE OF SHORT TERM INVESTMENT
FE	FCC	4	10	41	0	0	0	FACTORING EXPENSE
ARF	FACT	99	0	42	0	0	0	ACCOUNTS RECEIVABLE TO BE FACTORED
CR	RATE	99	0	43	0	0	0	CREDIT RATING
IE	BIN	4	10	44	0	0	0	INTEREST EXPENSE
TL	BANK	4	0	45	0	0	0	TERM LOAN BALANCE
TLIR	BANK	99	0	46	0	0	0	TERM LOAN INT RATE
STLOUT	OLB	4	10	47	0	0	0	OUTSTANDING SHORT TERM LOAN
STL	SHORT	4	0	48	0	0	0	SHORT TERM BORROWING
STIR	SR	99	0	49	0	0	0	SHORT TERM INTEREST RATE
SPL	OL	99	0	50	0	0	0	OUTSTANDING SPECIAL LOANS
SV	SL	4	10	51	2	2	4	SALES VOLUME IN UNITS(PROD,AREA)
PRICE	PR	0	0	55	2	2	4	PRICE(PROD,AREA)
MS	PCT	2	0	59	2	2	4	MARKET SHARE(PROD,AREA)
ADV	AD	4	0	63	2	2	4	ADVERTISING BUDGET(PROD,AREA)
SREPS	OAGT	0	0	67	2	0	3	NUMBER OF SALES REPRESENTATIVES(AREA)
TRAINEES	TRAIN	99	0	69	0	0	0	CURRENT TRAINEES
SC	COM	4	10	70	0	0	0	TOTAL SALES SALARIES AND COMMISSIONS, CURRENT PERIOD
SAL	CDOL	1	10	71	0	0	0	SALARY FOR SALESMEN
COMM	CPER	1	0	72	2	0	4	SALES COMMISSION(PROD)

LS	SLL	99	0	74	2	2	4	LOST SALES DUE TO STOCK-OUT(PROD,AREA)
SHIP	KSP	99	0	78	2	0	4	FLAG TO INDICATE EMERGENCY SHIPPING PERMIT(PROD)
QC	QC	4	10	80	0	0	0	QUALITY CONTROL BUDGET, CURRENT PERIOD
QC2%	PQC	4	20	81	0	0	0	% OF QC BUDGET WHICH GOES TO PRODUCT 2
RD	RD	4	20	82	2	0	4	PRODUCT R&D(PROD)??????
PVFS	PV	4	20	84	2	2	4	PRODUCTION VOLUME, 1ST SHIFT (PROD,AREA)
PC	CY	3	20	88	2	2	6	PLANT CAPACITY IN MAN-HOURS(STAGE,AREA)
CPC	DLTP	3	20	92	2	2	6	CHANGE IN PRODUCTION CAPACITY(STAGE,AREA)*
MAINT	PX	4	20	96	2	2	6	PLANT MAINTENANCE EXPENDITURE(STAGE,AREA)
FGI	EV	4	10	100	2	2	4	FINISHED GOODS INVENTORY(PROD,AREA)
BO	BACKO	99	0	104	2	2	4	BACKORDERS(PROD,AREA)
RMI	RINV	4	20	108	2	2	5	RAW MATERIAL INVENTORIES(MATERIAL,AREA)
RMO	RAW	99	0	112	2	2	5	RAW MATERIAL ORDERS(MATERIAL,AREA)
UTX	TRANS	99	0	116	2	2	4	UNITS TRANSHIPPED(PROD,AREA)
ICC	ECG	4	10	120	0	0	0	INVENTORY CARRYING CHARGES
RMUC	RMP	4	20	121	2	2	5	RAW MATERIAL UNIT COST
FGUC	STDC	4	10	125	2	2	4	FINISHED GOODS UNIT COST
TPC	PCST	4	20	129	2	2	4	TOTAL PRODUCTION COSTS(PROD,AREA)
PSB	ENG	5	20	133	0	0	0	PROCESS STUDY BUDGET
CRF	CRF	99	0	134	0	0	0	COST REDUCTION FACTOR
SS%	XTC	99	0	135	2	0	3	% OF FIRST SHIFT SCHEDULED FOR SECOND SHIFT
PVSS	PV2	4	0	137	2	2	4	PRODUCTION VOLUME, 2ND SHIFT (PROD,AREA)
SUM		0	0	-1	0	0	0	SUM FUNCTION
MAX		0	0	-2	0	0	0	MAXIMUM FUNCTION
MIN		0	0	-3	0	0	0	MINIMUM FUNCTION
AVE		0	0	-4	0	0	0	AVERAGE FUNCTION
LOG		0	0	-5	0	0	0	LOG FUNCTION

ENDCOMLISPRISUBOPTLP;TERBATREWREMWHEXXX

ABCDEFGHIJKLMNOPQRSTUVWXYZ*/+-()=,; .<>0123456789

83
13
127
37
33
57
5

AC	4	10	235	0	0	0
ADV	4	0	276	2	2	4
AP	4	10	227	0	0	0
ARF	99	0	216	0	0	0
AR	4	10	210	0	0	0
AVE	0	0	161	0	0	0
BBI	0	0	164	0	0	0
BL	0	0	171	0	0	0
BOUT	0	0	180	0	0	0
BO	99	0	252	2	2	4
CASH	4	10	167	0	0	0
CE	99	0	140	0	0	0
COGS	4	10	145	0	0	0
COMM	1	0	196	2	0	4
CPC	3	20	279	2	2	6
CRF	99	0	224	0	0	0
CR	99	0	126	0	0	0
DEPR	4	0	104	0	0	0
DIVD	0	0	108	0	0	0
EINQ	0	0	71	0	0	0
EINY	0	0	65	0	0	0
EI	0	0	56	0	0	0
FC	99	0	57	0	0	0
FE	4	10	94	0	0	0
FGI	4	10	182	2	2	4
FGUC	4	10	200	2	2	4
FUTL	0	0	44	2	0	5
FUT	0	0	232	2	0	5
FVBO	0	0	232	0	0	0
ICC	4	10	324	0	0	0
IE	4	10	241	0	0	0
IT	99	0	211	0	0	0
LOG	0	0	164	0	0	0
LS	99	0	247	2	2	4
MAIN	4	20	337	2	2	6
MAX	0	0	146	0	0	0
ME	4	10	172	0	0	0
MIN	0	0	131	0	0	0
MS	2	0	204	2	2	4
OE	4	0	150	0	0	0
PC	3	20	286	2	2	6
PIC	4	0	138	0	0	0
PLAN	4	20	148	0	0	0
PRAW	0	0	116	2	0	5
PRIC	0	0	156	2	2	4
PSB	5	20	236	0	0	0

```
PVFS                4      20      235    2  2    4
PVSS                4       0      221    2  2    4
QC2%                4      20      162    0  0    0
QC                  4      10      144    0  0    0
QTR                 0       0       44    0  0    0
RD                  4      20      156    2  0    4
RMI                 4      20      220    2  2    5
RMO                99       0      148    2  2    5
RMUC                4      20      219    2  2    5
SALE                4       0      219    0  0    0
SAL                 1      10      276    0  0    0
SC                  4      10      271    0  0    0
SHIP               99       0      267    2  0    4
SOFF                0       0      203    0  0    0
SOUT                0       0      195    0  0    0
SPL                99       0      204    0  0    0
SP                  0       0      185    0  0    0
SREP                0       0      223    2  0    3
SSP                 0       0      170    0  0    0
SSR                 0       0      162    0  0    0
SS%                99       0      265    2  0    3
STBR                0       0      130    0  0    0
STII                0      10      158    0  0    0
STIR               99       0      147    0  0    0
STI                99       0      130    0  0    0
STLO                4      10      150    0  0    0
STL                 4       0      134    0  0    0
SUM                 0       0       74    0  0    0
SV                  4      10      169    2  2    4
TAX                 4       0       83    0  0    0
TLIR               99       0       95    0  0    0
TL                  4       0       96    0  0    0
TPC                 4      20      249    2  2    4
TRAI               99       0       97    0  0    0
UBD                 0       0       63    0  0    0
UTX                99       0      146    2  2    4
YEAR                0       0      207    0  0    0
ENDCOMLISPRISUBOPTLP;TERBATREWREMWHEXXX
ABCDEFGHIJKLMNOPQRSTUVWXYZ*/+-()=,;  .<>0123456789
IACC1=      243
IACC2=      200
IACC3=      292
IACC4=      135
IACC5=      119
IACC6=      197
IACTOT=    2388
```

Appendix E
Sample Course Outlines

MSci 6055
MANAGEMENT INFORMATION SYSTEMS

The objective of this course is to teach the conceptual design of information systems from a managerial perspective.

Although the course has a managerial orientation, a considerable amount of technical material will be covered. This is necessary because a solid foundation in both computer systems and managerial concepts is required for one to be able to design information systems.

You can expect to spend a considerable amount of time at the computer terminals, since much of the work will involve "hands-on" use.

Also, you will participate in playing a management game. To do this the class will be organized into teams. Your team will play the game and will also be required to design and implement an information system to help you make decisions. The team project can be expected to require considerable time and effort, and will constitute a major portion of the work load.

DETAILS:

TEXTS: Burch and Strater, Information Systems: Theory and Practice, Prentice-Hall, 1977.

Jensen and Cherrington, The Business Management Laboratory, Business Publications, Inc., 1977.

GRADING:
Mid-term Exam 1.....20%
Mid-term Exam 2.....20%
Final Exam.....20%
Term Project.....40%
100%

SYLLABUS: A syllabus has been stored on a file under the instructor's computer account. To get a listing of the syllabus, use the following commands:

```
GET,RNF/UN=LIBRARY.
GET,MS6055/UN=IMJFC**.
ASCII.
RFL,40000.
REDUCE(-)
```

```
RNF,MS6055,OUTPUT.
```

Before entering the last line, advance to the end of the page, so your output will start on a new page. Also, if you wish to have your output sent to a file other than OUTPUT, substitute the file name for OUTPUT in the last line above.

	SUBJECT	ASSIGNMENT
June 23	MIS Introduction, BML	BML Manual Text Chpt. 1-2
28	Systems Theory	Text Chpt. 3 Mgt. Systems readings (lib.) BML Decision 1
30	MIS Structural Design	Text Chpt. 4-5 BML Decision 2
July 3	Case Study: Automated Accounting and Mgt. Info System (AAMIS) for Engineering Experiment Station	Report on reserve in library BML Decision 3
7	Systems Analysis	Text Chpt. 10-11 BML Decision 4
10	Detailed Systems Design	Text Chpt. 12-13 BML Decision 5
14	Exam 1	
17	Introduction to SLIM query language	SLIM manual BML Decision 6
21	SLIM query language (cont'd)	BML Decision 7
24	File Organization	Text Chpt. 8 BML Decision 8
26	File Organization (cont'd)	BML Decision 9
31	Data Structures	Text Chpt. 9 BML Decision 10
Aug. 4	Data Structures	BML Decision 11
7	Systems Implementation	Text Chpt. 14 BML Decision 12
11	AAMIS: Part II	BML Decision 13
14	Exam 2	
17	GDBMS - System 2000	BML Decision 14
21	Team MIS presentations	
25	Team MIS presentations	

BUSINESS 292
MANAGEMENT DECISION LABORATORY

Scope of course:

This course integrate the concepts of management decisions, planning and policies through the use of a computer executed simulation. Students participate in teams to manage a simulated business over several time periods, stressing rationality in business decisions and integrating functional fields into management decisions. Open only to last quarter business school students, or by permission of the instructor.

Course objectives:

1. To provide an appreciation for and understanding of the need to integrate the management of the functional areas to achieve overall company success.

2. To provide an appreciation for and understanding of the problems of decision making under conditions of uncertainty and competition with limited information.

3. To improve analytical skills through practice on an on-going problem situation.

4. To improve oral and written communication skills through class presentations and written assignments.

5. To introduce the student to the managerial use of computer based decision support systems via an experiential process.

Evaluation:

Participation will be evaluated through several measures:

1.	Team results, performance of the simulated firm		30%
2.	Peer evaluation, written evaluation of each member of the team		10%
3.	Annual meeting: Annual report	10%	
	Presentation	10%	20%
4.	Other reports and assignments as given		40%

Several assignments will take the form of in-class problems which require written answers or recommendations. You are expected to

be present for these on the same basis as if they were examinations. These may include hand-out readings or other reference assignments, may require in-class written work, or may involve other forms of response. These are the "I walked into the office this morning and this is the problem that landed on my desk.." type assignment. They are shown as CMTBA on the assignment schedule.

CLASS	DATE	ASSIGNMENT
1.	3/27	Introduction, team selection begins
2.	3/30	Rules, prepare dummy decision, introduce SLIM
3.	4/3	Discussion of dummy run, Q&A, introduce EMPACT
4.	4/6	Decision 1, Q&A session
5.	4/10	Each team introduces themselves and their firm to the 'analysts society' meeting
6.	4/13	Decision 2, more discussion on SLIM
7.	4/17	Decision 3, written team assignment. (as new managers, prepare a statement of your firm's written policies and how they are to guide your actions. Cover all functional areas.)
8.	4/24	Decision 4. Meet with individual firms.
9.	4/27	Decision 5. Meet with individual firms.
10.	5/1	Decision 6, CMTBA
11.	5/4	Decision 7, CMTBA
12.	5/8	Decision 8, CMTBA
13.	5/11	Decision 9, CMTBA
14.	5/15	Decision 10, Meet with individual firms.
15.	5/18	Decision 11, Meet with individual firms.
16.	5/22	Decision 12, CMTBA
17.	5/25	Decision 13, Annual Report due. Plan a presentation to your board of directors. Assume a fiscal year ending with decision 12. Be prepared to discuss your handling of the problems encountered and relationships to your company policies.
18.	5/29	CMTBA
19.	6/1	Wrap-up discussion, peer evaluations due, other in-class items will be assigned at class time.